5-31-1990

Non-parametric dominant point detection algorithms

Kuo-wei Huang
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

Title of Thesis :

Non-parametric Dominant Point

Detection Algorithms.

Kuo-wei Huang, New Jersey Institute of Technology,
Master of Electrical Engineering, 1990

Thesis directed by : *Dr. Nirwan Ansari*

New Jersey Institute of Technology,
Department of Electrical and
Computer Engineering Center for
Communications and Signal Processing.

Dominant point detection is one of the most important preprocessing steps for landmark-based shape recognition and point-based motion estimation. Two new methods for detecting dominant points are presented. Both methods do not require any input parameter and the dominant points obtained by these methods remain relatively the same even when the object curve is scaled or rotated.

In the first method, for each boundary point, a support region is assigned to the point based on its local properties. Each point is then smoothed by a Gaussian filter with a width proportional to its determined support region. A significance measure for each point is then computed. Dominant points are finally obtained through nonmaximum suppression.

The second method is rather simple. It traces the contour of an object curve, and, for each boundary point, it assigns a "chain code" which indicates the direction of the trace. Dominant points are then determined by detecting the change of direction of each point.

These two methods lead to an important observation that the performance of a dominant points detection algorithm depends not only on the significance measure and the support region but also on the presence of noise.

Unlike other dominant point detection algorithms which are sensitive to scaling and rotation of the object curve, these two new methods will overcome this difficulty. Furthermore, they are robust in the presence of noise.

The proposed two methods are compared to those of several other dominant point detection algorithms in terms of the CPU processing time, the approximation errors and the number of the detected dominant points of a given curve.

Non-parametric Dominant Point Detection Algorithms

by

Kuo-wei Huang

Thesis submitted to the Faculty of the Graduate School of the New Jersey Institute of Technology in partial fulfillment of the requirements for the degree of Master of Sience in Electrical Engineering, 1990

APPROVAL SHEET

Title of Thesis :

Non-Parametric Dominant Point Detection Algorithms

Name of Candidate : *Kuo-wei Huang*

Student of Electrical Engineering, 1990

Thesis and Abstract Approved :

5/11/90
Date

Signature of other members of the thesis committe :

5/11/90
Date

5/11/90
Date

VITA

Name : *Kuo-wei Huang.*

Permanent address :

Degree and date to be conferred : M.S.E.E.,1990

Date of birth :

Place of birth :

Secondary education :

Collegiate institutions attended	Dates	Degree	Date of Degree
Min Chi Institute of Technology	Sep. 1978	Diploma	May 1983
New Jersey Institute of Technology	Sep. 1988	Master	May 1990

Major : Electrical Engineering

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to my advisor, Dr. Nirwan Ansari, for his guidance, encouragement throughout this research effort.

I wish to thank my wife, Ming-Ying Ching for her support and patience, and my fellow graduate students in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology for their friendship.

Furthermore, I would like to express my gratitude to other Committee members, Dr. Ali N Akansu and Dr. Irving Wang, for serving on the examination committee and for evaluatuing this research.

TABLE OF CONTENTS

LIST OF FIGURES:

Chapter 1

Introduction

It has been suggested from the viewpoint of the human visual system [3] that dominant points along an object contour are rich in information content and are sufficient to characterize the object contour. The dominant points are the high curvature points along a digital curve that have important shape attributes.

Many algorithms have been suggested for detecting dominant points. They fall into two categories [2][11][12][16][20][23][25][29] — one is to find the dominant points through edge or angle detection and the other is to obtain a piecewise linear polygonal approximation of the digital curve subject to certain constraints on the goodness of fit.

In polygonal approximation, dominant points are the intersecting points of any two adjacent line segments. These points are also known as the vertices or break points of the closed curve(polygon).

Due to the discrete boundary representation and quantization errors, false local concavities and convexities along a boundary are introduced when estimating curvature. Smoothing is thus necessary to reduce those false concavities and convexities, it has been shown in [22] that a Gaussian filter is an ideal smoothing filter for numerical differentiation.

Most dominant-point detection algorithm(either angle detection or polygonal approximation), except [28], require one or more input parameters. According to [29], these parameters usually represent the region of support for the measurement of local properties(curvature). In general , it is hard, if not impossible, to find a set of parameters that will work on a curve with various size or rotation. Too large a region of support will smooth out the fine features of a curve, whereas too small a region of support will generate a large number of redundant dominant points. This difficulty can be avoided by a non-parametric method [29] in which the support region of each contour point is obtained based on its local properties.

In this thesis, we introduce two new methods to detect dominant points.

The first method adopts the idea of support region [29], and follows by Gaussian smoothing [3]. The smoothing is used to suppress noise. The details of this method will be discussed in Chapter 3. The second method obtains dominant points by detecting the change of direction of the contour tracing. It is a simple and fast method, and performs as well as any other methods without the presence of noise. This method will further be discussed in Chapter 4.

In Chapter 2, we will review other dominant points detection algorithms, and the elements of detecting dominant points. The review includes the Gaussian smoothing algorithm [30], the angle detection procedure by Rosenfeld and Johnston [23], the improved angle detection procedure by Rosenfeld and Weszka [27], the Determination of region of support by Teh-Chin algorithm [29], the Freeman chain code [26], the Freeman-Davis corner finding algorithm [12], the Sankar-Sharma dominant-point detection procedure [28], the Anderson-Bezdek vertex detection algorithm [1].

In Chapter 5, a comparison among the two new methods and other previously known algorithms will be made in terms of the maximum error, the integral squares error, the computational complexity and the number of detected dominant points. We will also discuss the effect and presence of noise

to the dominant point detection problem.

Chapter 2

Literature Review and The Elements for Detecting Dominant Points

2.1 Introduction

In this chapter, various dominant point detection algorithms will be reviewed briefly, and the elements used in this thesis, such as, Gaussian smoothing and the determination of support region, will be discussed.

2.2 Review

In this section, several dominant points detection algorithms will be reviewed briefly. The review includes the angle detection algorithm by Rosenfeld-Jothnston [23], the improved angle detection by Rosenfeld-Weszka [27], the corner finding algorithm by Freeman-Davis [12], the Sankar-Sharma dominant-point detection procedure [28] and Anderson-Bezdek vertex detection algorithm [1].

2.2.1 Rosenfeld-Johnston Angle Detection Procedure

This parallel procedure [23] is analogous to the Rosenfeld-Thurston edge detection algorithm [14] which detects significant maxima in average gray-level gradients by using a variable degree of smoothing. This procedure will be described in the next section. However, this algorithm can lead to incorrect results when edges occur too close to one another [26]. The results of detecting dominant points by this procedure are shown in Appendix A.

2.2.2 The Improved Angle Detection Procedure

This procedure [27] is modified from the above Rosenfeld-Johnston algorithm [23]. It was originally developed by Davis and Rosenfeld [10], and was later improved by Rosenfeld and Weszka [27]. Although, it has overcome the difficulty described above, it still needs some input parameters. The results of dominant points detected by this algorithm are shown in Appendix A.

2.2.3 Freeman-Davis Corner Finding Algorithm

This algorithm [12] starts with scanning the chain $\{c_i, i = 1,...,n\}$ with a moving straight line segment which connects the end points of a sequence of s links (it is defined as $\overline{c_i c_{i+1}}$). As the line segment moves from one chain node to the next, the angular differences between successive segment positions are used as a smoothed measure of local curvature along the chain.

This algorithm also exhibits the same problem of the Rosenfeld-Johnston algorithm [23]. It also need some input parameters, and when the object curve is round-shape, a big error will happen. The results of detecting dominant points by this algorithm are shown in Appendix A.

2.2.4 Sankar-Sharma Dominant-point Detection Procedure

In this procedure [28], the dominant points are computed iteratively as the points of maximum global curvature, based on the local curvature of each point with respect to its immediate neighbors. First it is observed that each point of a closed curve having exactly two neighbors can be classified into three classes based on the local curvature, as shown in Fig. 2.1

Type of Curvature	Weight Assigned
No curvature	0
Positive curvature	+1
Negative curvature	-1

Fig. 2.1 Local curvature assignment.

For those points having more than two neighbors, let (i, j) be the point with k immediate neighbors where $k \geq 3$. Find all possible pairs of 2-neighbor configurations of (i, j). Assign to each such pair the corresponding local curvature, using Fig. 2.1. From this collection, omit those pairs which

have been assigned positive (negative) curvature, then the point (i, j) is assigned the weight $+1(-1)$; on the other hand, if some pairs have positive curvature while the rest have negative curvature, then the point (i, j) is assigned the weight 0.

Because no smoothing procedure is included in this algorithm, false local concavities and convexities along a boundary are introduced. The results of dominant points detected by this algorithm are shown in Appendix A.

2.2.5 Anderson-Bezdek Vertex Detection

In this algorithm [1], tangential deflection and curvature of discrete curve are defined based on the geometrical and statistical properties associated with the eigenvalue-eigenvector structure of sample covariance matrices.

Specifically, it has been proven [1] that the nonzero entry of the commutator of a pair of scatter matrices constructed from discrete arcs is related to the angle between their eigenspaces, and the entry is also proportional to the analytical curvature of the plane curve from which the discrete data are drawn.

The sequential algorithm then identifies the location of vertices of the

discrete curves based on excessive cumulative tangential deflection between successive vertices, an approach which differs markedly from all the previous approaches of searching for points of relative curvature extreme.

The results of dominant points detected by this algorithm are shown in Appendix A. This algorithm has the same problem as the Rosenfeld-Johnston algorithm and it also needs input parameters. If the object curve has sharpen-angle shape, some real dominant points will be missed.

2.3 Background

In this section, elements for detecting dominant points that will be used in this thesis are discussed.

2.3.1 Gaussian Smoothing Algorithm

As mentioned before, a Gaussian filter is an ideal smoothing filter for numerical differentiation [3]. Using this filter to smooth an object contour, some noise points present on the contour can be eliminated. Note that the encoded Gaussian smoothed curve emphasizes the flatness of the region between ridges. A planar curve [3] can be defined in a parametric form ,

$$(x(t), y(t)) \in R^2 \tag{2.1}$$

where t is the path length along the curve.

Basically, smoothing by a Gaussian filter is just convoluting $x(t)$ and $y(t)$, respectively,. with a Gaussian filter. A one-dimensional Gaussian filter can be written as follow ,

$$\eta(t,\omega) = \frac{1}{\sqrt{2\pi}\omega} exp^{\frac{-t^2}{2\omega^2}} \tag{2.2}$$

where ω is the width (spatial support) of the filter.

The smoothed curve is denoted by the set of points:

$$(X(t,\omega), Y(t,\omega)). \tag{2.3}$$

where

$$X(t,\omega) = x(t) * \eta(t,\omega), \tag{2.4}$$

$$Y(t,\omega) = y(t) * \eta(t,\omega), \tag{2.5}$$

and $*$ indicates the convolution operator.

It can be shown that the Gaussian smooth curvature [3] is :

$$k(t,\omega) = \frac{\dot{X}\ddot{Y} - \dot{Y}\ddot{X}}{(\dot{X}^2 + \dot{Y}^2)^{3/2}} \tag{2.6}$$

where

t is the path length along the curve.

ω is the width of the Gaussian filter.

k is the curvature at t.

and

$$\dot{X} = \frac{dX}{dt}, \ddot{X} = \frac{d^2X}{dt^2} \tag{2.7}$$

$$\dot{Y} = \frac{dY}{dt}, \ddot{Y} = \frac{d^2Y}{dt^2} \tag{2.8}$$

Note that tracing the object curve along the contour in increasing values

of t, a positive curvature corresponds to a concavity on the left, and a negative curvature corresponds to a concavity on the right.

2.3.2 Freeman Chain Code

A digital closed curve C can be denoted by a sequence of n integer-coordinate points [12],

$$C = \{p_i = (x_i, y_i), i = 1, ..., n\} \tag{2.9}$$

where p_{i+1} is a neighbor of p_i(module n).

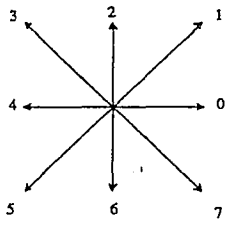


Fig. 2.2 Freeman code.

The Freeman chain code of C consists of n vectors, written as $\{c_i, i = 0, 1, 2, ..., n-1\}$, where

$$c_i = \overline{p_{i-1}p_i}, \tag{2.10}$$

and c_i takes on an integer f where

$$f = \{0, ..., 7\} \tag{2.11}$$

as shown in Fig. 2.2, Note that $f/4\pi$ is the angle between the X-axis and the vector. The chain of C is defined by $\{\vec{c_i}, i = 1, ..., n\}$ and $\vec{c_i} = \overline{c_{i\pm n}}$. All integers are of module n [29].

2.3.3 Rosenfeld-Johnston Procedure

On the real Euclidean plane, the curvature of a curve is defined as the rate of change of slope as a function of arc length. For a curve $y = f(x)$ [27], it can be expressed as follows :

$$\frac{\frac{d^2y}{dx^2}}{[1 + (\frac{dy}{dx})^2]^{3/2}} \tag{2.12}$$

If the curvature at p_i is defined by simply replacing the derivatives

above by the first differences, the successive slope angles on the digital curve can only differ by a multiple of 45^o, and thus small changes in slope cannot be evaluated. This problem can be alleviated by using a smoothed slope measurement, e.g., defining the slope at p_i as,

$$\frac{(y_{i+k} - y_i)}{(x_{i+k} - x_i)}, \tag{2.13}$$

for some $k > 1$, rather than simply using the first difference ($k = 1$).

Define the k-vectors at p_i as

$$\overline{a_{ik}} = (x_i - x_{i+k}, y_i - y_{i+k}) \tag{2.14}$$

$$\overline{b_{ik}} = (x_i - x_{i-k}, y_i - y_{i-k}) \tag{2.15}$$

so the k-cosine at p_i is defined,

$$c_{ik} = \frac{\overline{a_{ik}} \cdot \overline{b_{ik}}}{\mid \overline{a_{ik}} \mid\mid \overline{b_{ik}} \mid} \tag{2.16}$$

where the c_{ik} is the cosine of the angle between the k vectors $\overline{a_{ik}}$ and $\overline{b_{ik}}$.

Thus, $-1 \leq c_{ik} \leq 1$, where c_{ik} is close to 1 if $\overline{a_{ik}}$ and $\overline{b_{ik}}$ make an angle close to 0^o, and c_{ik} is close to -1 if $\overline{a_{ik}}$ and $\overline{b_{ik}}$ make an angle near 180^o ; that is, c_{ik} is larger when the curve is turning rapidly , and smaller when the curve is relatively straight. To choose an appropriate k, we use the following procedure :

1. Select a smoothing factor m. For example, let m be $[n/10]$, i.e. 1/10 of the perimeter of the curve.

2. Assign region of support h and curvature value $c_{i,h}$ to point p_i for the largest h such that,

$$c_{i,m} < c_{i,m-1} < \ldots < c_{i,h} \geq c_{i,h-1} \tag{2.17}$$

   where c_{i0} = -1 and the cosine at p_i, is denoted by c_i.

3. Finally, retain those points p_i where $c_{i,h_i} \geq c_{j,h_j}$ for all j such that $\mid i - j \mid \leq h/2$.

2.3.4 Improved Angle Detection Algorithm

The above angle detection algorithm is analogous to the edge detection algorithm developed in [28]. It will obtain an incorrect result when edges occur too close to one another.

A modified scheme [25] overcomes this difficulty. In the modified angle detection scheme, the k-cosines are smoothed by averaging. Each point with $(k > 1)$ will be smoothed as follows :

$$\overline{c_{ik}} = \frac{2}{k+2} \sum_{j=k/2}^{k} c_{ij} \qquad for \quad k = even \qquad (2.18)$$

$$\overline{c_{ik}} = \frac{2}{k+3} \sum_{j=(k-1)/2}^{k} c_{ij} \qquad for \quad k = odd \qquad (2.19)$$

The $\overline{c_{ik}}$ are then treated just like the c_{ik} in the original method [23].

2.3.5 Teh-Chin Algorithm

In the Rosenfeld-Johnston procedure [23], the h_i is chosen to satisfy the following constraint :

$$c_{i,m} < c_{i,m-1} < \ldots < c_{i,hi} \geq c_{i,h_{i-1}}. \tag{2.20}$$

Note that h_i is the region of support. Davis [9] pointed out that incorrectly chosen regions of support may cause the measures of significance to be computed over inappropriate neighborhoods which may subsequently cause dominant points to be discarded. In the Rosenfeld-Johnston procedure, the h_i is decided from an input parameter.

To overcome the difficulty of choosing incorrect regions of support, Teh and Chin [29] use the following procedure :

1. Determine the length of the chord joining the points $\overline{p_{i-k}p_{i+k}}$ as,

$$l_{ik} = \mid \overline{p_{i-k}p_{i+k}} \mid . \tag{2.21}$$

Let d_{ik} be the perpendicular distance from the point p_i to the chord $\overline{p_{i-k}p_{i+k}}$.

2. Start with $k = 1$. Compute l_{ik} and d_{ik} until,

(a) $l_{ik} \geq l_{i,k+1}$

or

$$(b) \quad \frac{d_{ik}}{l_{ik}} \geq \frac{d_{i,k+1}}{l_{i,k+1}} \qquad for \quad d_{ik} > 0 \qquad (2.22)$$

$$\frac{d_{ik}}{l_{ik}} \leq \frac{d_{i,k+1}}{l_{i,k+1}} \qquad for \quad d_{ik} < 0 \qquad (2.23)$$

Then the region of support of p_i is the set of points which satisfy either condition (a) or condition (b),

$$D(p_i) = \{(p_{i-k}, ..., p_{i-1}, p_i, p_{i+1}, ..., p_{i+k}) \mid condition \ (a) \ or \ condition \ (b)\} \qquad (2.24)$$

2.3.6 The Computation of Perpendicular Distance

Let p_{i-1},p_i and p_{i+1} be three points in the 2-Dimensional ($2D$) plane. To compute the perpendicular distance from p_i to the chord joining p_{i-1} and p_{i+1}, we use the following procedure [6] :

1. The area of this triangle made by p_{i-1}, p_i and p_{i+1} is computed as follows :

$$S_1 = p_i - p_{i-1}. \tag{2.25}$$

$$S_2 = p_i - p_{i+1}. \tag{2.26}$$

$$S_3 = p_{i+1} - p_{i-1}. \tag{2.27}$$

$$P = (S_1 + S_2 + S_3)/2. \tag{2.28}$$

$$A = \sqrt{(P \times (P - S_1) \times (P - S_2) \times (P - S_3))} \tag{2.29}$$

Where S_1, S_2, S_3 is the three side lengths of this triangle.

- P is the parameter of the triangle.
- A is the area of the triangle.
- The perpendicular distance D is, thus,

$$D = \frac{(2 \times A)}{S_3}. \tag{2.30}$$

Chapter 3

The First Method

3.1 Introduction

In this Chapter, the Teh-Chin dominant points detection procedure and the polygonal approximation procedure will be discussed in details. The advantages and disadvantages of these two procedures are analyzed, and the implementation of first method will be described.

3.2 Motivation

Image processing is concerned with the manipulation and analysis of picture by computer [2]. Its major subareas include:

1. Acquisition and compression : Converting analog pictures to digital form; efficient coding or approximation of pictures so as to save memory space.

2. Enhancement, restoration and reconstruction : Improving degraded pictures; reconstructing pictures by integrating partial information such as projections.

3. Matching, description, and recognition : Comparing and registering pictures to one another; segmenting pictures into parts, measuring properties and relationships among the parts, also comparing the resulting descriptions to models that define classes of pictures.

This thesis will focus on detecting dominant points, a necessary segmentation procedure to achieve landmark-based shape recognition [3]. As mentioned before, many dominant point detection algorithms have been suggested, and they can be divided into two categories, edge or angle detection

and polygonal approximation.

In the former category, the dominant points are obtained by measuring the angles(curvature) of the boundary points. The angle is related to the curvature, and the smaller the angle, the larger the curvature. There are many method for computing the curvature. The following three are adopted in the Teh-Chin algorithm :

1. The cosine measure in Rosenfeld-Johnston procedure, c_{ik} (This has been discussed in Chapter 2).

2. Curvature measure : The curvature at point p_i is taken to be the difference in the mean angular direction of k vectors on the leading and trailing curve segments of the point. It is as follows :

$$CUR_{ik} = \frac{1}{k}\sum_{j=-k}^{-1} f_{i-j} - \frac{1}{k}\sum_{j=0}^{k-1} f_{i-j} \tag{3.1}$$

where f_{i-j} is the Freeman chain code.

3. 1-curvature measure : By taking $k = 1$ for the curvature measure, the 1-curvature measure is defined by,

$$CUR_{i1} = f_{i+1} - f_i. \tag{3.2}$$

Each boundary point is characterized by its support regional curvature. Candidates for dominant points are boundary points which have the maximum curvature in their corresponding support regions. The dominant points are finally obtained from the candidate points by using various suppression schemes.

3.2.1 An Example of Detecting Dominant Point by Angle Detection

Teh-Chin algorithm [29] is a good example of detecting dominant points that falls in the first category, angle-detection. The algorithm does not require input parameter, and it makes use of the concept of support region. The dominant points detected by Teh-Chin algorithm is better than others in many aspects, such as the maximum error, the integral square error, the number of detected dominant points and the computational complexity. The algorithm works as follows :

1. Determine the region of support for each point. This procedure is described in Chapter 2,

$$D(p_i) = \{p_{i-k}, \ldots, p_{i-1}, p_i, p_{i+1}, \ldots, p_{i+k}\}. \tag{3.3}$$

2. Select a measure of significance(curvature) from one of the three defined significance measures mentioned above, and calculate its absolute value for each point, $\mid S(p_i) \mid$.

3. Suppress nonmaximum points.

   (a) 1st pass : Perform nonmaxima suppression as follows : retain only those points p_i where

$$\mid S(p_i) \mid \geq \mid S(p_j) \mid \tag{3.4}$$

for all j such that

$$\mid i - j \mid \leq \frac{k_i}{2}. \tag{3.5}$$

(b) 2nd pass : Further suppress those points having zero 1-curvature (CUR_{i1}=0).

(c) 3rd pass : For those points survived after 2nd pass, if ([k_i of D(p_i)] = 1) and (p_{i-1} or p_{i+1} still survived) then further suppress p_i, if ($\mid S(p_i) \mid \leq \mid S(p_{i-1}) \mid$) or ($\mid S(p_i) \mid \leq \mid S(p_{i+1}) \mid$).

If 1-curvature is selected as a measure of significance, then go to step (3d), otherwise those points survived are the dominant points.

(d) 4th pass : For those groups of more than 2 points that still survived, suppress all the points except the two end points of each of the groups.

Owing to quantization error and noise effect, the false concavities and convexities along object boundary are introduced. These false concavities and convexities can easily be detected as dominant points.

The above noise factor was not considered in Teh-Chin algorithm. If the object curve is corrupted with noise, noisy boundary points may have high

significance measure; thus, resulting in incorrect dominant points.

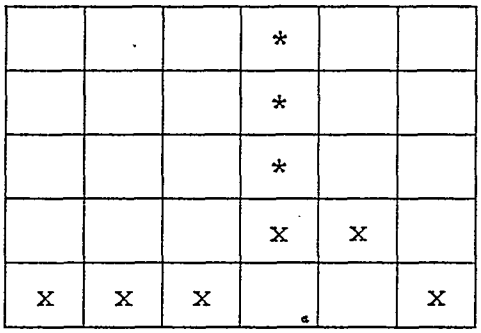
The Teh-Chin algorithm works well on object curve which is not corrupted with noise. However, it will produce false dominant points if the curve is noisy. We will demonstrate this disadvantage by an experimental result in Chapter 5. To reduce the noise effect, we introduce a smoothing procedure to be incorporated with the Teh-Chin algorithm. This smoothing procedure is used in "curvature guided polygonal approximation" method [3] which will be discussed next.

3.2.2 An Example of A Polygonal Approximation Algorithm

The "curvature guided polygonal approximation" [3] is a good example of detecting dominant points by polygonal approximation. Here, a preprocessing step, Gaussian smoothing, is used to reduce effect of noise and quantization error. The algorithm can be summarized as follows :

1. Remove all one-pixel wide protrusions. An example of a one-pixel wide protrusion which may result due to the discrete boundary representa-

tion and quantization error, shown in Fig. 3.1 :



* indicates a boundary protrusion pixel.

x indicates a boundary pixel.

Fig. 3.1

2. Smooth the boundary with a Gaussian filter.

3. Find the set of positive maximum and negative minimum curvature points on the smoothed boundary.

4. The points along the original boundary that correspond to the set of points found in Step (3) are used as the starting set of break points for polygonal approximation of the original boundary.

5. Employ the split-and-merge polygonal approximation algorithm. A split-and-merge polygonal approximation procedure is summarized as follows :

   (a) Assign an arbitrary number of points along the boundary as the initial set of break points. The initial approximated polygon is formed by joining the sequence of break points along the original boundary with straight lines.

   (b) For each pair of adjacent break points, determine the point along the boundary portion that yields the maximum perpendicular distance to the straight line segment joined by the two break points. If the maximum perpendicular distance is greater than the given tolerance, that point becomes a new break point; i.e., the line segment is replaced

by two line segments. This is the "splitting" part of the algorithm.

(c) For each pair of adjacent line segments consisting of three consecutive break points, say A, B and C, compute the maximum perpendicular distance from the boundary portion between A and C to line $\overline{AC}$.

If the distance is in the tolerance, break point B is removed. That is, line segment $\overline{AB}$ and $\overline{BC}$ are replaced by line segment $\overline{AC}$. Note that each replacement is immediately tested for merging with the next line segment. This is the "merging" part of the algorithm.

(d) Replace Steps (b) and (c) until an equilibrium is reached; i.e., no more splitting and merging.

6. The resulting break points are the dominant points of the boundary.

Though this algorithm reduces the effects of noise, it requires two parameters, namely, the width, ω, of the Gaussian filter and the collinearity

tolerance. There is a trade-off in selecting the width , ω. That is, a larger value of ω will remove small details of the boundary curvature, while a smaller value will permit false concavities and convexities. It is very hard to chose an appropriate ω.

3.2.3 Evaluation

While the Teh-Chin algorithm is non-parametric, it is sensitive to noise. On the other hand, the "curvature guide polygonal approximation" reduce the effect of noise, but it requires two input parameters. By taking the advantages of both algorithms, we introduce a new method which is both non-parametric and less sensitive to noise. The method employs the concept of support region and Gaussian smoothing.

If noise is not severe, we also introduce a very computational efficient method which works as good as the Teh-Chin algorithm. This method is simply done by detecting change of direction from the Freeman chain code. The first method will be described in details in the next section. The second method will be discussed in the next chapter.

3.3 The First Method

The first method that we introduce can be summarized as follows :

1. Contour tracing : Encode the contour by Freeman chain code.
2. Determine the support region for each contour point by the Teh-Chin algorithm. This has been discussed in Section 2.2.5.
3. Smooth the contour by a Gaussian filter with a width proportional to the support region.
4. Compute significance measure for each point on the Gaussian smoothed curve. Here, we used the cosine measure(Equation 2.16) as our significance measure.
5. According to the significance measure obtained in Step (4), do the nonmaximum and nonminimum suppression.

We will discuss Steps (3) and (5) of the algorithm in the following sections.

3.3.1 Selection of the Width of A Gaussian Filter

In this section, we will discuss the relationship between the support region and the width of the gaussian filter. It has been shown that a Gaussian

filter is an ideal smoothing filter for numerical differentiation [30]. For convenience, we will repeat part of the discussion in Section 2.2.1. Denote a planar curve by a set of points in parametric form,

$$(x(t), y(t)) \in R^2 \tag{3.6}$$

where t is the path length along the curve.

Smoothing the curve with a Gaussian filter is equivalent to convoluting $x(t)$ and $y(t)$, respectively, with a one-dimensional Gaussian filter,

$$\eta(t, \omega) = \frac{1}{\sqrt{2\pi}\omega} exp^{\frac{-t^2}{2\omega^2}} \tag{3.7}$$

where ω is the width(spatial support) of the filter.

For a given support region, what should ω be ?

According to the support region determination algorithm(Equation 2.21 to 2.25), the joining chord length $\overline{p_{i-1}p_{i+1}}$ of p_i, and the perpendicular distance d_i(from p_i to $\overline{p_{i-1}p_{i+1}}$) are used as parameters in determining the support region. In a closed digital curve, the curvature of each boundary point

is affected by and related to its neighbor points. A break point is formed at the intersection of two segments of a consecutive sequence of points where points on one segment have positive(negative) slopes while points on the other segment have negative(positive) slopes.

Thus, the definition of support region for a contour point p_i can be stated as,

- An expanded region on both side of p_i. Within this region, the point p_i should have the maximum curvature than other points.

When noise is present on the contour, we use a Gaussian filter to smooth the contour to reduce the noise effect. Again, since concavities and convexities along an object contour occur at different scales, how much smoothing is necessary ? In addition, as mentioned earlier, the curvature of a point is determined by its neighboring points, and thus, we want to make the amount of smoothing adaptive.

Since the measurement of the curvature of a point should be based on the local properties within its region of support, we propose to smooth the point with a Gaussian filter of length equal to the region of support. What should be the width of the filter ? Intuitively, the neighboring points closer

to the point of interest should have higher weights than those points further away.

Figures 3.2 to 3.10 (shown in the end of this chapter), show the shape of a Gaussian filter with widths = 1, 1.5, 2, 2.5, 3.3, 4, 4.5, 5 and 5.5, respectively. The window length of the filter is 21 which can be considered to be 2(support region) + 1.

From the figures, with too large a width, points that are far away from the point of interest are contributing too much to the smoothing while with too small a width, they hardly contribute at all. The best compromise, subjectively, is when ω=3.3 as shown in Fig. 3.6 where the neighboring points closer to the point of interest have higher weights while points farther away still contributes the smoothing but in less amount. In conclusion, each point is smoothed by a Gaussian filter of "length = 2(support region) + 1" and with "width = 0.33 $\times$ (support region)."

The above analysis can be used to explain the trade-off in selecting the width ω of a Gaussian filter : "A larger value of ω will remove small details of the boundary curvature, a smaller value of ω will permit false concavities and convexities."

The Gaussian smoothing is done as follows :

1. For each boundary point, determine its support region. The support region is determined by the two conditions which are mentioned in Section 2.2.5 (Equation 2.22, 2.23 and 2.24). However, for the special case when a contour point forms a straight line with its two adjacent points, we make the support region of this point "0."

2. Determine the width ω of the Gaussian filter. That is, $\omega = 0.33 \times$ support region.

3. Use a Gaussian filter with length = 2(support region) + 1, and width obtained in Step (2).

4. Each smoothed point is thus obtained by the weighted sum of the products of the boundary points within the region of support and the respective coefficients of the Gaussian filter.

   Two special condition should be noted,

   - For a point with a support region equal to 0, skip the smoothing step. Because the point with a "0" support region, should be a point along a straight line. In this situation, its curvature is equal to -1 (the smallest curvature), and therefore it cannot be a

dominant point.

- For the point with a support region equal to 1, we use the following smoothing procedure :

$$(\hat{x}_i, \hat{y}_i) = 0.4(x_i, y_i) + \frac{1}{4}[(x_{i-1}, y_{i-1}) + (x_{i+1}, y_{i+1})] \qquad (3.8)$$

This is because the point with support region equal to 1, is usually a point characterized by a region with small details. Thus, this point should be smoothed out. If we use Gaussian smoothing with the length and the width defined in Steps (2) and (3), this point will be immuned from smoothing. Therefore, we smooth this point according to Equation (3.8).

3.3.2 Determination of Dominant Points

Based on the computed significance measure of each contour point, the last step is to determine which of those contour points are the dominant points.

In the Teh-Chin algorithm [29], four levels of suppression are used to chose the dominant points. In the first level suppression, any contour point

that have the maximum significance measure within half of its support region is eliminated as a dominant point. In the second level, any point having a zero 1-curvature (defined by Equation 3.2) is suppressed. In the third level suppression, if there are two consecutive points that survive from the first and second level suppression, eliminate the one with less significance.

In the "curvature guided polygonal approximation algorithm," points with local maximum and minimum curvature are first selected. That is, all nonmaximum and nonminimum curvature points are suppressed. However, a "split and merge polygonal approximation ," is then used to alternatively selecting and suppressing the possible dominant points.

Strategy for selecting dominant points

The first method we introduced use the following two levels of suppression :

1. Select the points that achieve the maximum significance measure within the corresponding support regions : For each point, check if this point achieves the maximum significance measure within its support region. If not, suppress this point.

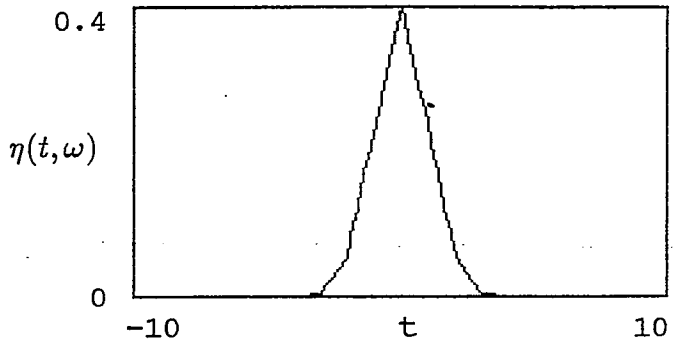
2. For those points survived after first level of suppression, if the support region of point p_i is equal to 1, and either p_{i+1} or p_{i-1} is still survived, then suppress the point with a smaller significance.

If small details along the object curve is desirable, such scheme will, however, eliminate the small details because of the Gaussian smoothing. The experimental results will be discussed in Chapter 5.

3.3.3 Summary

We have reviewed the Teh-Chin algorithm and "curvature guide polygonal approximation " in great details. Their advantages and disadvantages are also discussed.

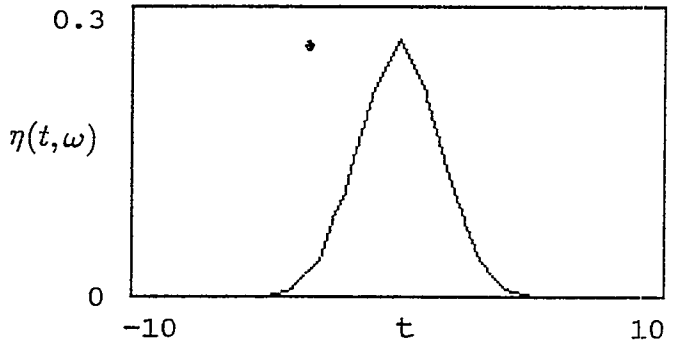
Our first method is introduced by incorporating the above two algorithms. It inherits the merits of both algorithms; that is, it is non-parametric and it is relatively robust in the presence of noise.



w := 1.

t := -10 ..10

Fig. 3.2.



w := 1.5

t := -10 ..10

Fig. 3.3.

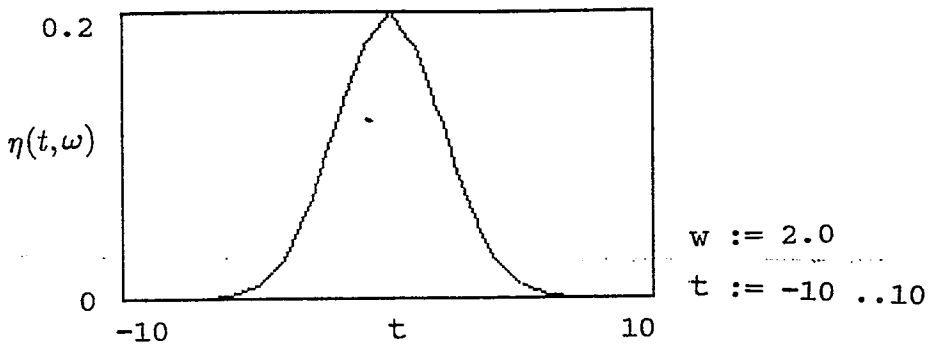


Fig. 3.4.

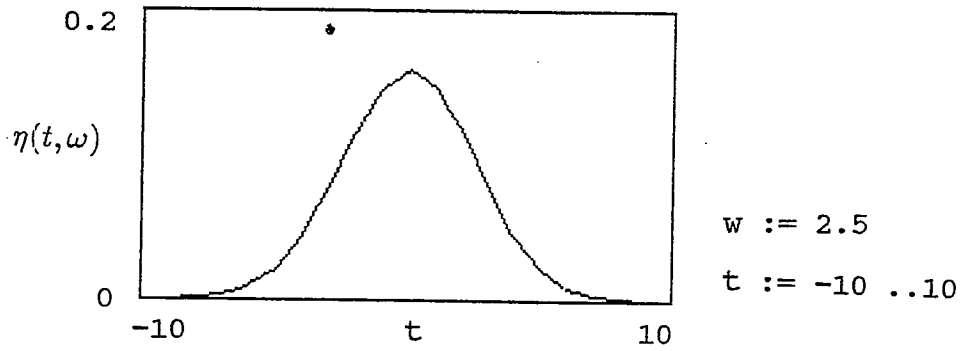


Fig. 3.5.

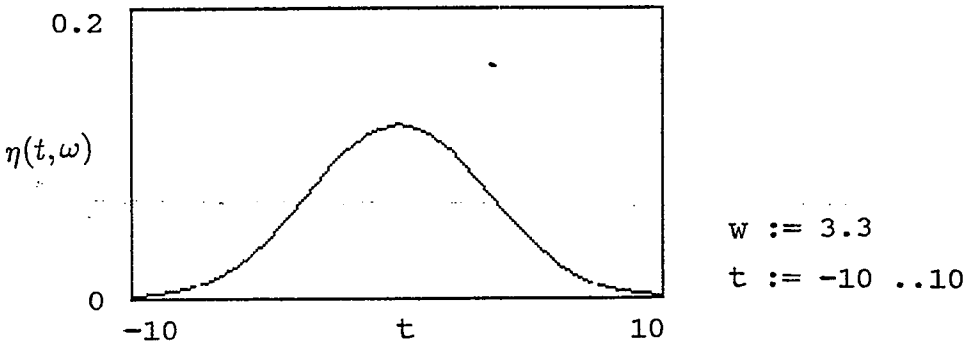


Fig. 3.6.

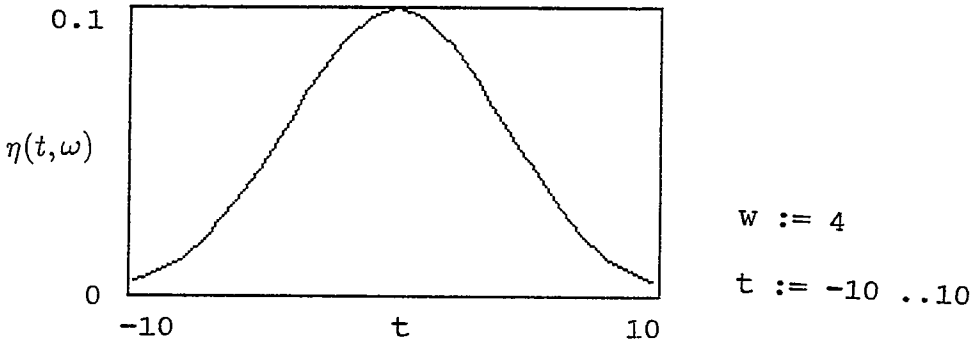
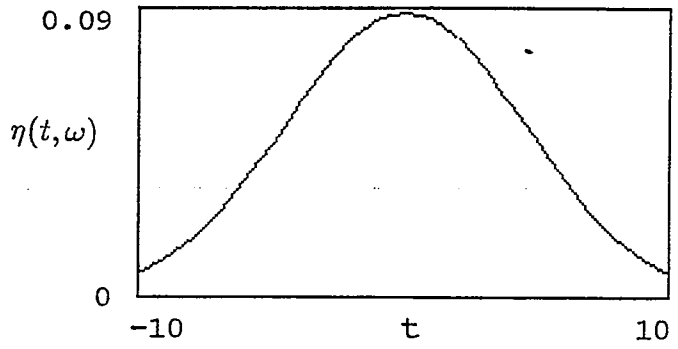


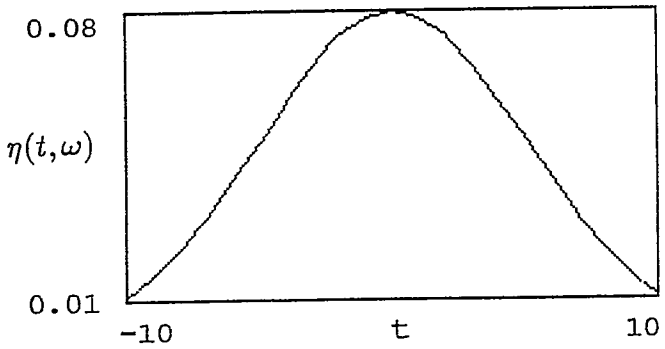
Fig. 3.7.



w := 4.5

t := -10 ..10

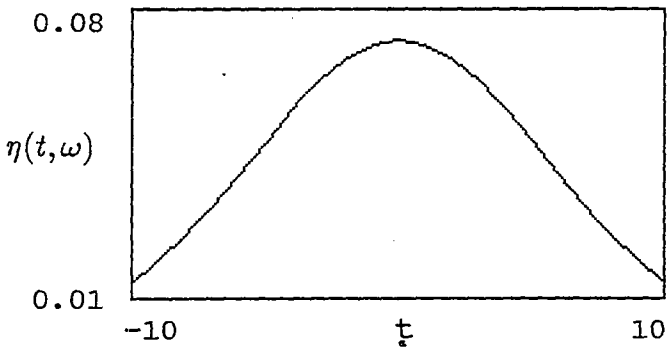
Fig. 3.8.



w := 5

t := -10 ..10

Fig. 3.9.



w := 5.5

t := -10 ..10

Chapter 4

The Second Method

4.1 Introduction

In this Chapter, we introduce a simple and computational efficient method which performs as well as the Teh-Chin algorithm in the noise-free case.

4.2 Motivation

The contour presented in the Teh-Chin algorithm [29] are all traced from the Freeman chain code. They are also noise-free. Under this situation, the vertices are usually the dominant points. Using the first method we

introduced in Chapter 3 to detect dominant points of this kind of object contour will result in missing dominant points because of the smoothing effect. Therefore, the second method is introduced.

4.3 The Second Method

If noise is not present on the object curve, detecting the dominant points on this curve is essentially detecting the vertices.

To trace an object curve by a Freeman chain code, each point on the contour will be assigned a digit number according to the direction of the trace of this point to the next adjacent boundary point. As mentioned in section 2.2.2, there are eight possible directions associated with the Freeman code. Each direction is represented by a digit number from 0 to 7.

The second method use Freeman chain code to trace the contour of an object curve. During the tracing, when the direction of the point p_i is different with the point p_{i+1}, then keep the point p_i as an dominant point. The dominant points obtained by this method are also the vertices of the object curve. Because it is based on the assumption of noise-free situation, the suppression scheme is not needed. Results of detecting dominant points by the

second method will also be discussed in Chapter 5.

4.4 Summary

According to above discussion, the second method is summarized by the following :

1. Encode the object contour by a Freeman chain code.

2. Check the Freeman code of each point, if the Freeman code of p_i is different with the Freeman code of p_{i+1}, that is, the trace for the next point has changed direction, p_i is considered as the dominant point.

Chapter 5

Conclusion

5.1 Introduction

In this Chapter, we present experimental results of detecting dominant points by the two methods we introduced. The results are compared to those obtained by other methods in term of the maximum error, the integral square error and the computational complexity. These three figure of merits have been used in many dominant point detection algorithms [23][25][26][29].

5.2 Experiment Results

The first and second dominant point detection methods have introduced and discussed in details in Chapter 3 and Chapter 4, respectively. Before we make any comparison among various dominant point detection algorithms, we want to emphasize two points – what are the dominant points used for ? and what kind of dominant points are concerned ?

As mentioned in the previous chapters, dominant points are used for shape recognition. The goodness of a dominant point detector thus directly depends on how the dominant points obtained by the detector can be readily used to achieve recognition.

For a given contour, we have to address the following three questions :

1. How many dominant points are sufficient to characterize the contour ?

   An adequate number of dominant points along an object contour are necessary to describe the object contour. If only a few number of dominant points can be detected, they may not be sufficient to describe the shape of the contour. However, if too many dominant points are detected, some of these points may be extraneous due to discrete representation of the object contour and noise. A bad dominant point

detector will burden the subsequent shape recognition algorithm to identify the shape of the object. Thus, the "adequate" number of dominant points to characterize an object contour is subjective, and closely depend on the subsequent postprocessing procedure.

2. Are the dominant points obtained by the detector invariant to the size and the orientation of the contour ? That is, for a given contour at a different scale and/or orientation, do the number of dominant points and their relative locations remain the same ?

3. Can the noise present on the contour be removed ?

   In the presence of noise, many false concavities and convexities along an object contour are introduced. A robust dominant point detector should be able to eliminate these false concavities and convexities as dominant points.

A robust dominant points detector is essential for shape recognition, we will use the above criteria to compare various dominant points detection algorithms.

In this thesis, we will experiment on the contours which are commonly used in the Teh-Chin algorithm [29] and many other algorithms [9][12][23].

According to the chain codes provided in [29], the original contour are shown in Fig. 5.1 to Fig. 5.4, namely and respectively, CHROMOSOME, LEAF, EIGHT and SEMICIR.

The results [29] of dominant points obtained various algorithms are shown in Figures A.1 to A.4, in Appendix A. Fig. 5.5 to 5.8 show the dominant points obtained from the set of contours (Fig. 5.1 to 5.4) by the Teh-Chin algorithm. The dominant points obtained by Teh-Chin algorithm on a set of rotated (by 30°) and scaled (by 2) contours are shown in Figures 5.9 to 5.16, respectively. Note that the rotated and scaled contours are superimposed with their respective resulting dominant points in the figures. The dominant points obtained is quite stable to scaling and rotation. The number of dominant points detected by this algorithm on the set of contours is about 25 % of total contour points. Note that the number of dominant points obtained are certainly dependent on the type of contours we are approximating.

5.2.1 The First Method

The results of detecting the dominant points on the contours shown in Figures 5.1 to 5.4 by using first method are shown in Figures 5.17 to 5.20.

The results for the rotated and scaled contours using the first method are shown in Fig. 5.21 to 5.28. Again, in each figure, we have superimposed the original contour with the resulting approximated contour. Like the Teh-Chin algorithm, this method is also stable to scaling and rotation. The number of dominant points obtained for the same given contour is also about 25 % of total contour points.

5.2.2 The Second Method

The second method simply uses the Freeman code to detect the dominant points. Support region, Gaussian smoothing and complicated significance measurement are not needed to determine the dominant points.

The results of the dominant points obtained by this method on the set of contours shown in Fig. 5.1 to 5.4 and their respective rotated and scaled contours are shown in Fig. 5.29 to 5.40. Since vertices of an object contour are considered as the dominant points, the method is naturally stable to rotation and scaling. However, the number of dominant points obtained is the largest because every vertices are considered as dominant points.

5.3 Comparison

In this section, the two new dominant point detecting methods are compared to other algorithms. Like other authors, the results are compared in term of the approximation errors, the number of detected dominant points, and the computational complexity.

- The Approximation errors :

In this thesis, we adapted the same quantitative measure of the quality of the detected dominant points described in the Teh-Chin algorithm [29]. The error between a point p_i of a digital closed curve C and the approximating polygon is defined as the perpendicular distance of the point to the approximating line segment. This error is denoted by e_i. Two error norms between C and its approximating polygon defined below are used :

1. Integral square error,

$$E_2 = \sum_{i=1}^{n} e_i^2 \tag{5.1}$$

2. Maximum error,

$$E\infty = max \quad e_i \tag{5.2}$$

The comparison results are shown in Table 5.1 to Table 5.4, respectively. In this thesis, the results of the Rosenfeld-Johnston algorithm, the Rosenfeld-Weszka algorithm, the Freeman-Davis algorithm, the Sankar-Sharma algorithm and the Anderson-Bezdek algorithm are directly obtained from [29]. The approximated contours detected by the above algorithms had been summarize by Teh and Chin [29]. Since they are not discussed in this thesis, the results are shown in Appendix B.

Though the Teh-Chin algorithm achieves very small error, the second method we introduce achieve the zero error. However, this does not mean that the second method is the best dominant point detector. Dominant point detector cannot be simply compared based on this quantitative measure of quality that are used by many researchers. The quality is usually very subjective. In general, large concavities and convexities should be dominant points. Small concavities and convexities may not be dominant points because these may be caused by noise and quantization error. If noise is not present, the second method would certainly be the best detector. If noise is severe, the

first method is preferable because this method will smooth out the effect of noise. We will emphasize this point by another experiment result later.

- The Computational Complexity.

The second method we introduce is the simplest and most computational efficient. The first method requires more computation than the Teh-Chin algorithm because additional computation is spent on Gaussian smoothing. In the next section, we will discuss the noise effect and justify that it is worth on taking additional computation to reduce the noise effect.

- The Effect of Noise

Consider a contour CROSS which has been corrupted with noise is shown in Fig. 5.41. The original contour should be a CROSS. The dominant points obtained by Teh-Chin algorithm are shown in Fig. 5.42, by the first method we introduce are shown in Fig. 5.43. Note that the contour approximated by the dominant points which are obtained by first method resembles curve closely to the original contour than that by the Teh-Chin algorithm. Since the Teh-Chin algorithm does not smooth out the noise, spurious points are taken as dominant points. The Gaussian smoothing incorporated in the first

method has gracefully reduce the noise effect, and thus eliminating most of the spurious points.

On the other hand, the second method would produce an approximated contour which is an exact copy of the corrupted CROSS contour.

5.4 Summary

We have introduced two new dominant point detection methods. The first method incorporates concept of the support region and Gaussian smoothing for detecting dominant points. The second method simply obtains the dominant points(or vertices) by detecting the change of direction along the contour being traced.

Both methods are relatively insensitive to the rotation and scaling of the object contour. The first method is robust in the presence of noise while the second method is very sensitive to noise. The goodness of a dominant point detector is very subjective . The figure of merits which have been used by many researcher to compare the quality of dominant point detectors, as points out, is not reliable. In the presence of noise, the first method performs better than the best known angle detection algorithm, the Teh-Chin

algorithm. If noise is not considered, the second method provides the best figure of merit, which, as we emphasize, are not good criteria for comparison.

TABLE 5.1

Results of the CHROMOSOME curve

Number of input Contour points n = 60				
Algorithm	No. of dominant points	Max. error	Integral error	CPU time (secs)
Rosenfeld and Johnston	8	1.54	21.94	4.19
Rosenfeld and Weszka	12	1.58	22.61	6.04
Freeman and Davis	8	1.51	22.56	3.12
Sankar and Sharma	12	2.03	28.89	16.46
Anderson and Bezdek	9	2.03	26.50	5.31
Teh-Chin by k-cosine	15	0.74	7.20	4.47
First Method	16	2.00	20.25	2
Second Method	37	0	0	1

TABLE 5.2

Results of the LEAF curve

Number of input Contour points n = 120				
Algorithm	No. of dominant points	Max. error	Integral error	CPU time (secs)
Rosenfeld and Johnston	17	1.76	43.42	10.91
Rosenfeld and Weszka	18	1.53	30.57	17.40
Freeman and Davis	17	1.72	45.27	8.16
Sankar and Sharma	20	3.48	71.15	44.63
Anderson and Bezdek	20	1.49	39.18	13.19
Teh-Chin by k-cosine	29	0.99	14.96	9.70
First Method	30	2.13	25.57	3
Second Method	57	0	0	1

TABLE 5.3

Results of the EIGHT curve

Number of input Contour points n = 45				
Algorithm	No. of dominant points	Max. error	Integral error	CPU time (secs)
Rosenfeld and Johnston	10	1.61	22.83	2.97
Rosenfeld and Weszka	16	1.59	12.67	3.74
Freeman and Davis	11	1.34	14.61	2.80
Sankar and Sharma	6	2.36	38.57	11.55
Anderson and Bezdek	9	1.11	8.97	4.49
Teh-Chin by k-cosine	13	1.00	5.93	3.99
First Method	14	1.00	5.06	2
Second Method	30	0	0	1

TABLE 5.4

Results of the SEMICIR curve

Number of input Contour points n = 102				
Algorithm	No. of dominant points	Max. error	Integral error	CPU time (secs)
Rosenfeld and Johnston	30	0.74	8.85	8.09
Rosenfeld and Weszka	34	1.00	15.40	10.13
Freeman and Davis	19	1.41	23.31	9.46
Sankar and Sharma	10	8.00	769.53	35.04
Anderson and Bezdek	29	1.18	6.43	12.04
Teh-Chin by k-cosine	22	1.00	20.61	9.66
First Method	28	1.26	17.83	2
Second Method	53	0	0	1

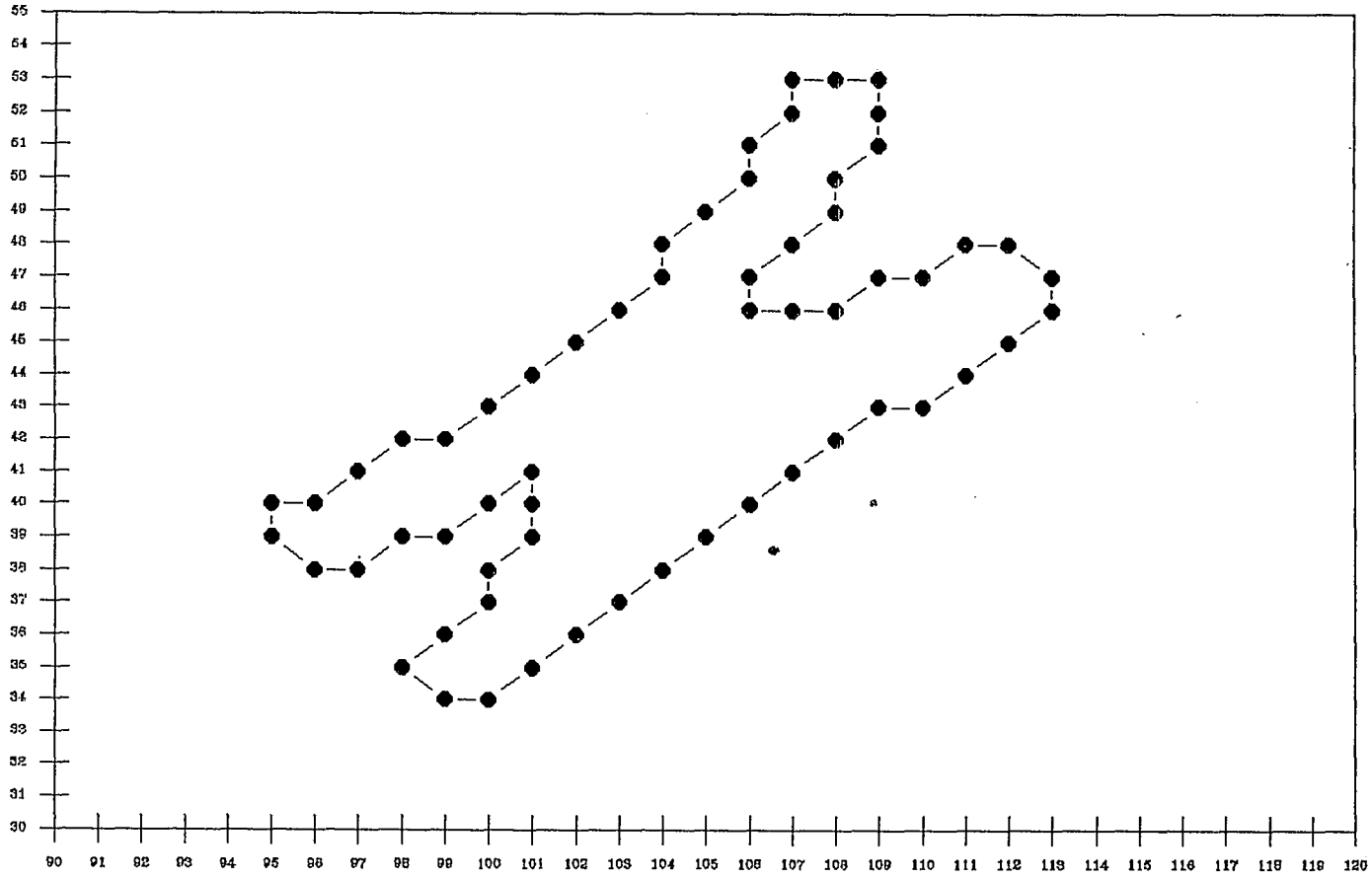


Fig. 5.1 The CHROMOSOME Contour.

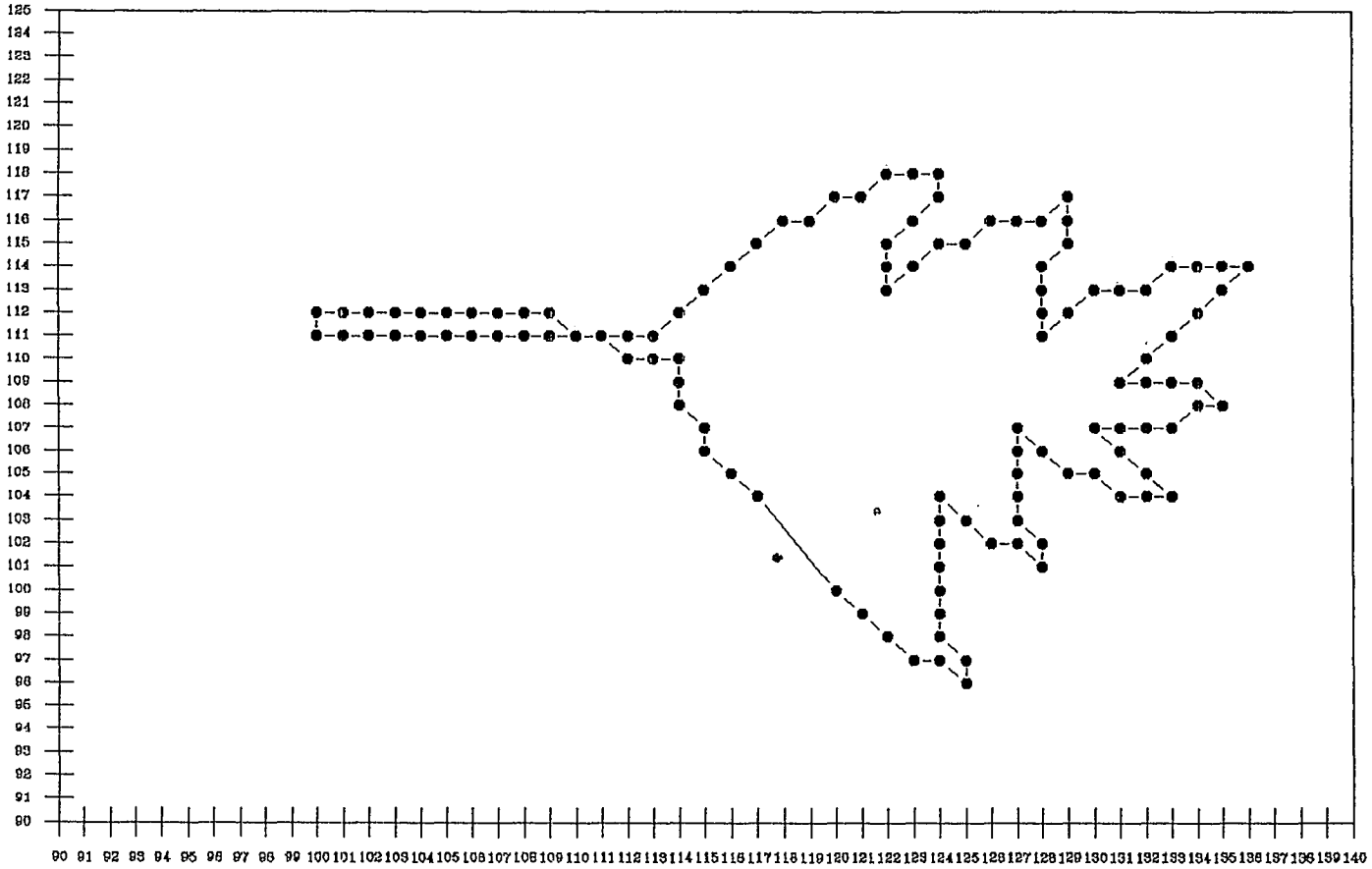


Fig. 5.2 The LEAF Contour

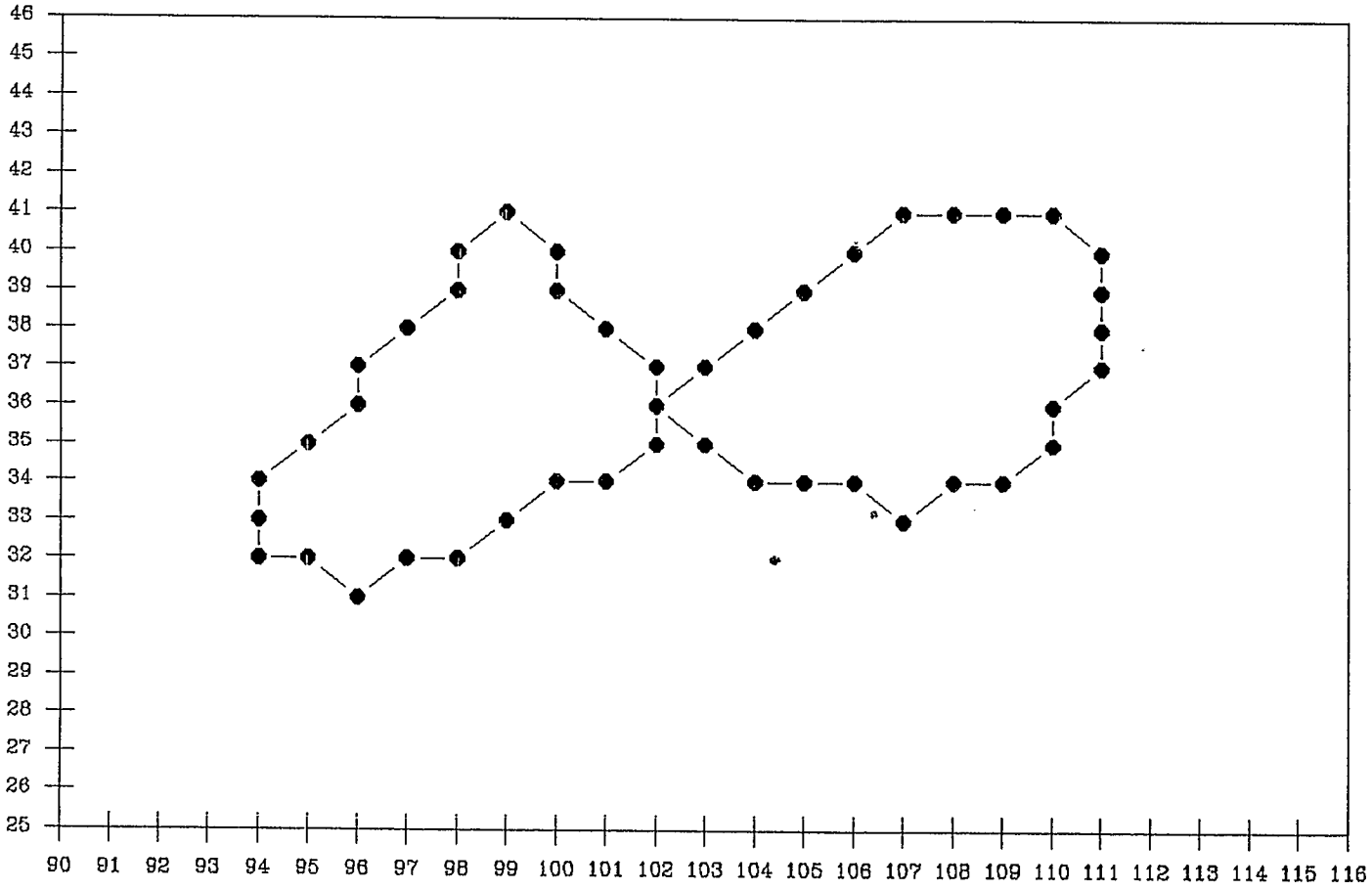


Fig. 5.3 The EIGHT Contour

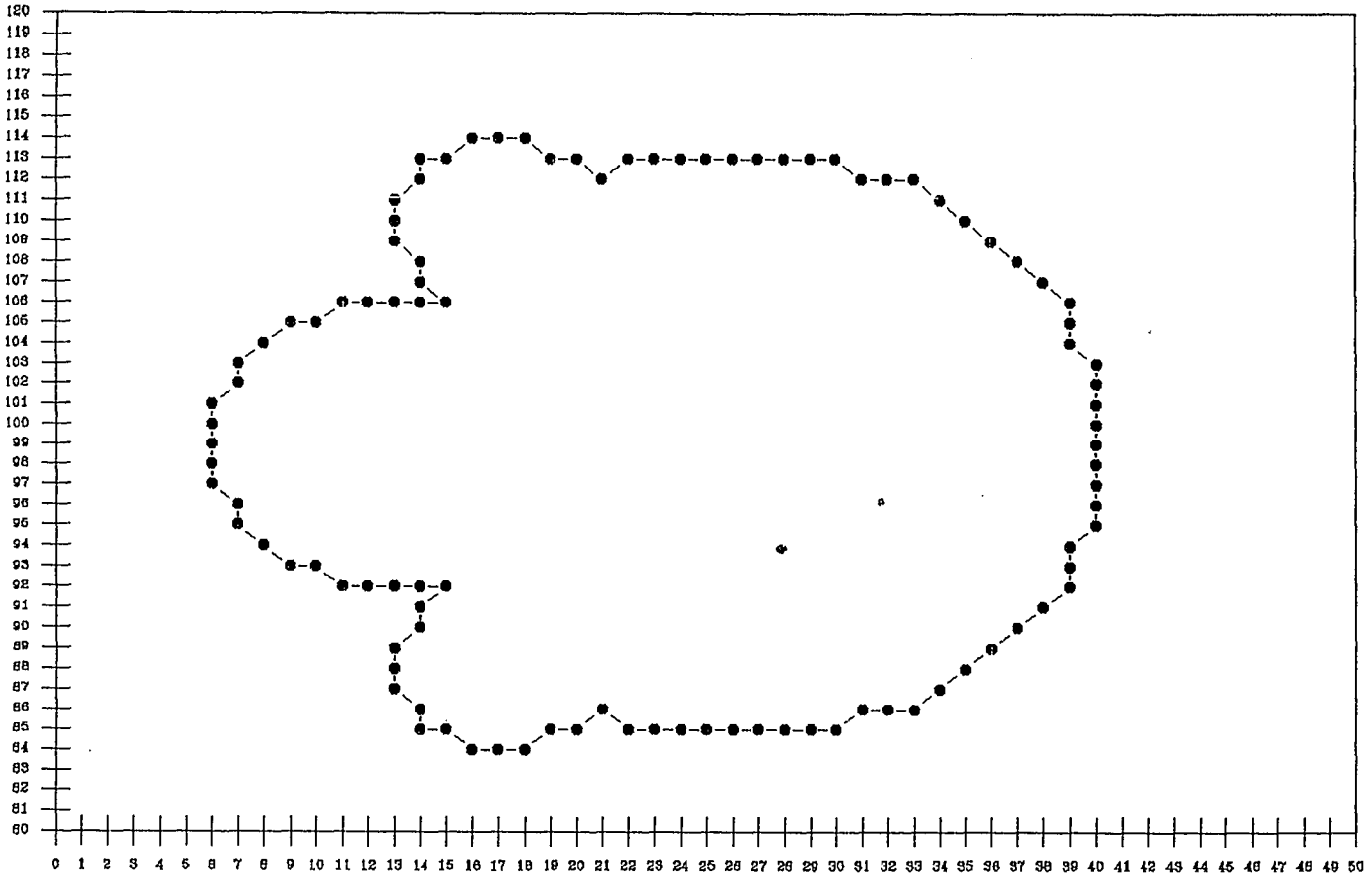


Fig. 5.4 The SEMICIR Contour

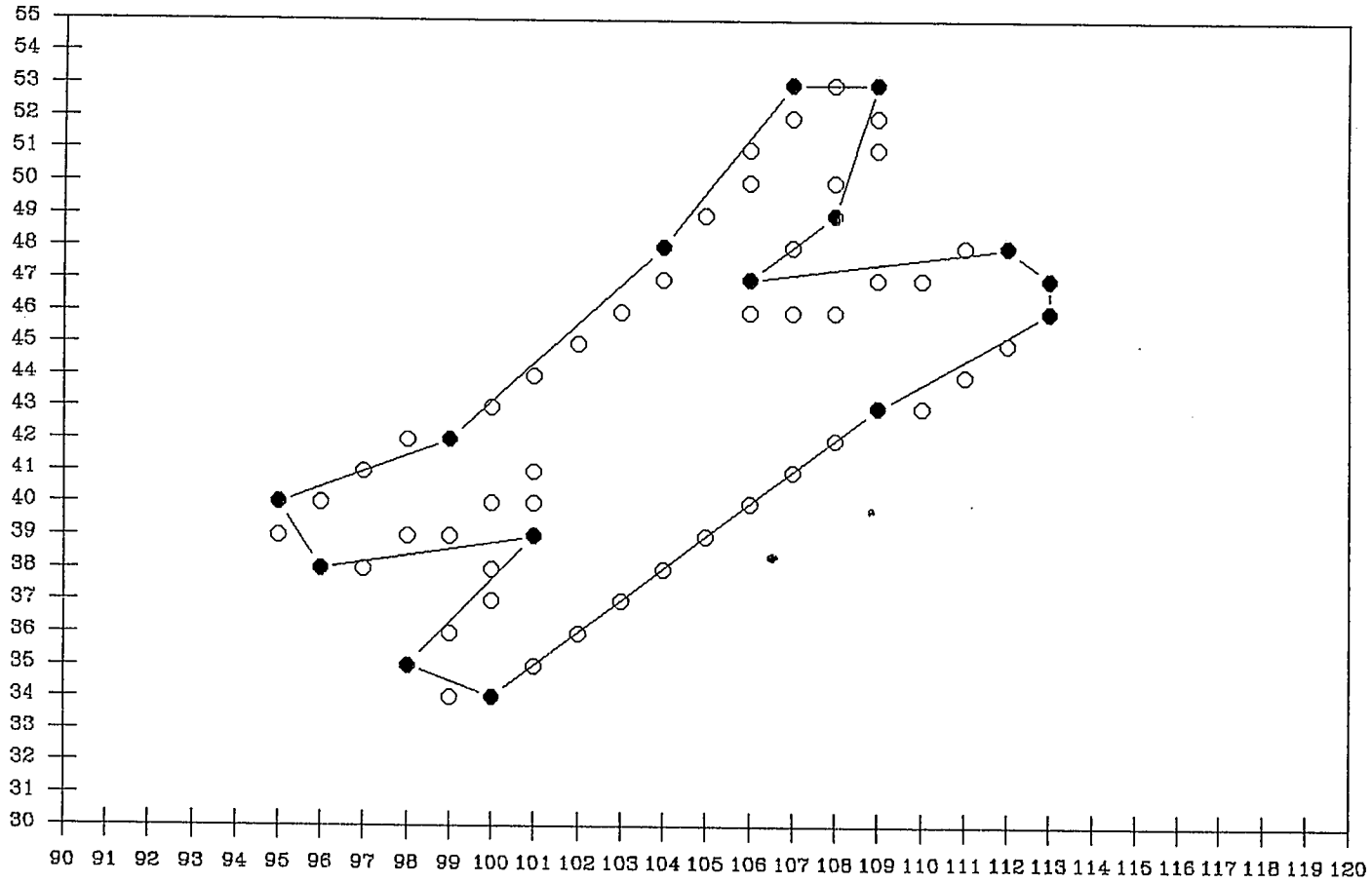


Fig. 5.5 Detected by Teh–Chin algorithm

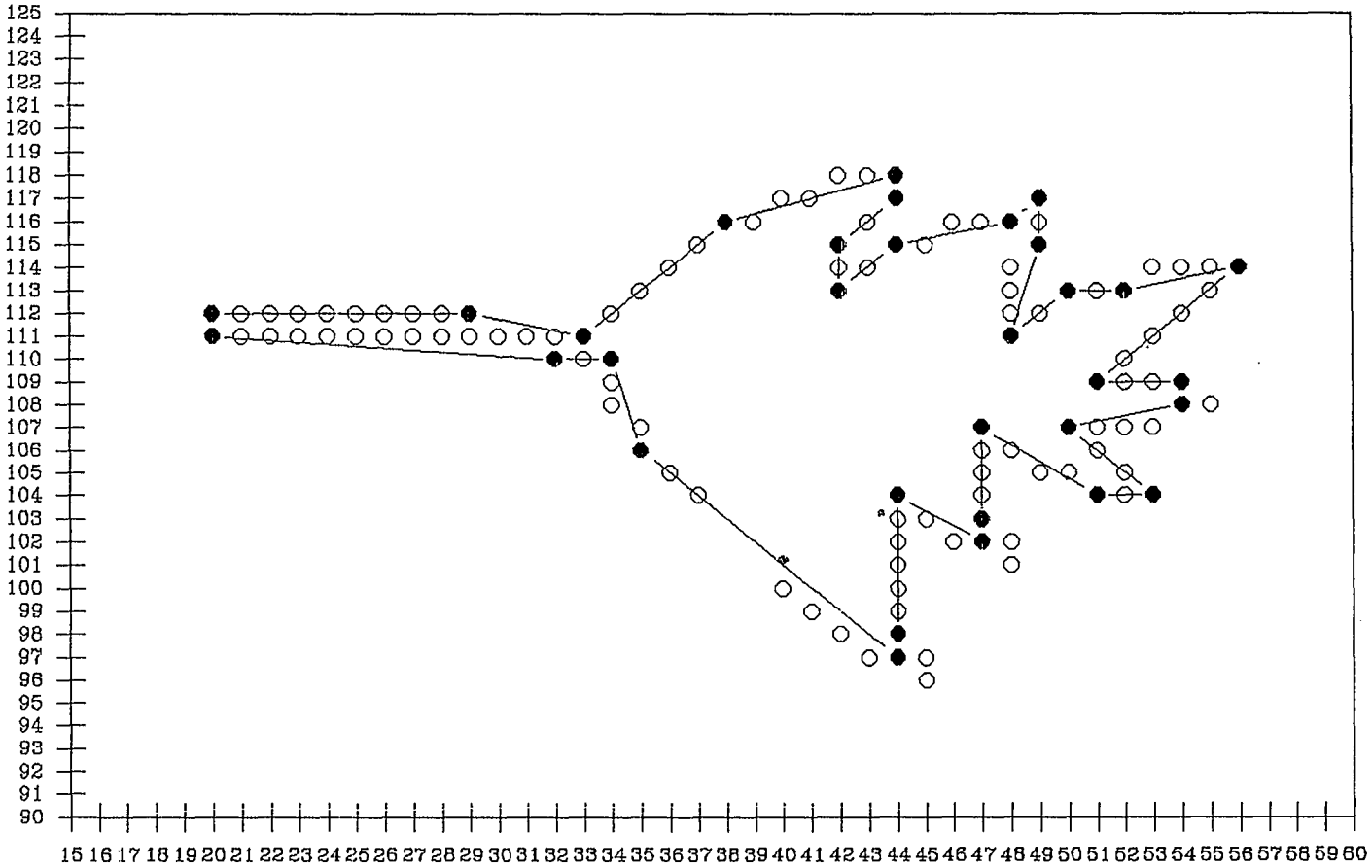


Fig. 5.6 Detected by Teh–Chin Algorithm

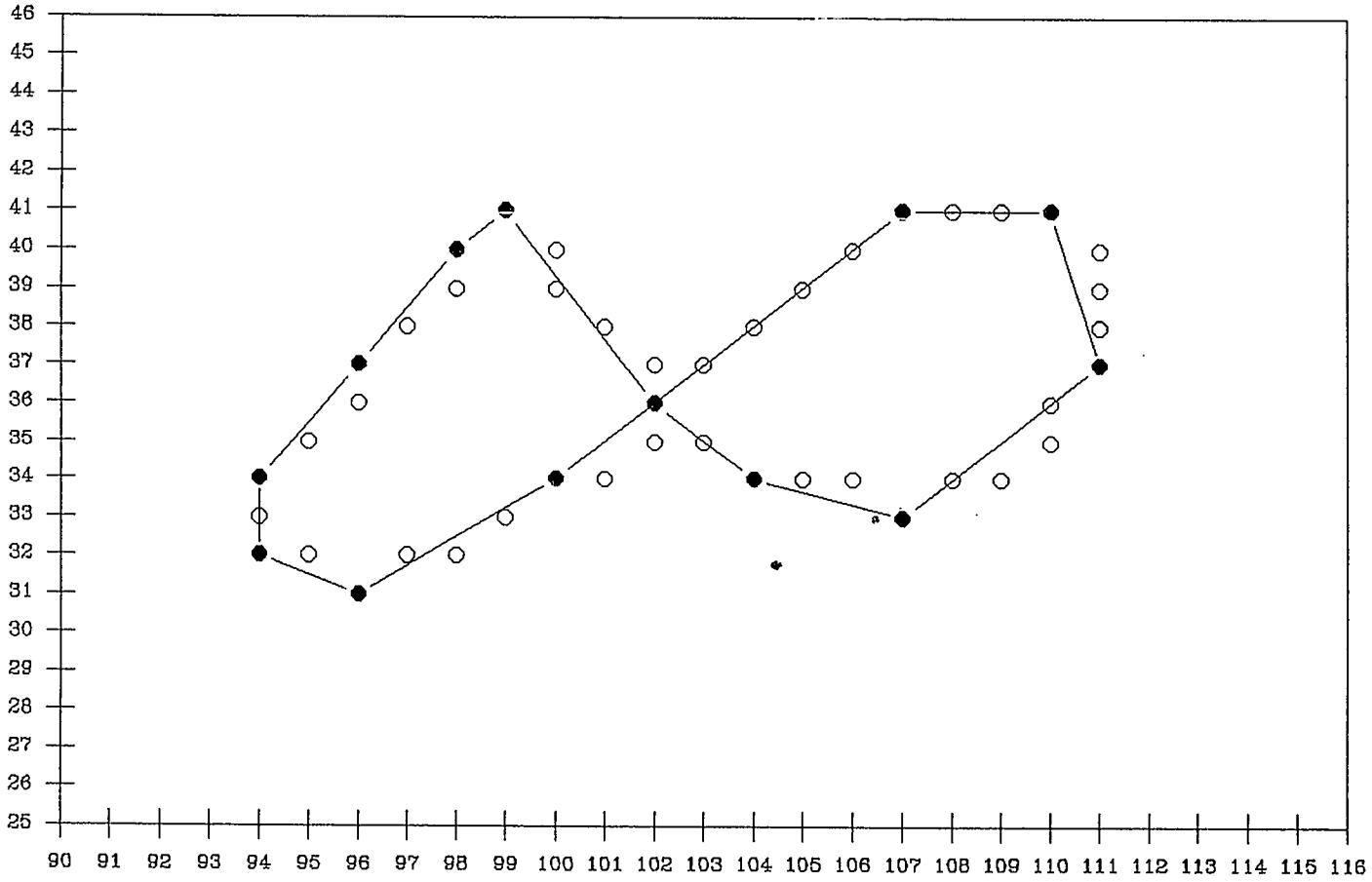


Fig. 5.7 Detected by Teh–Chin Algorithm

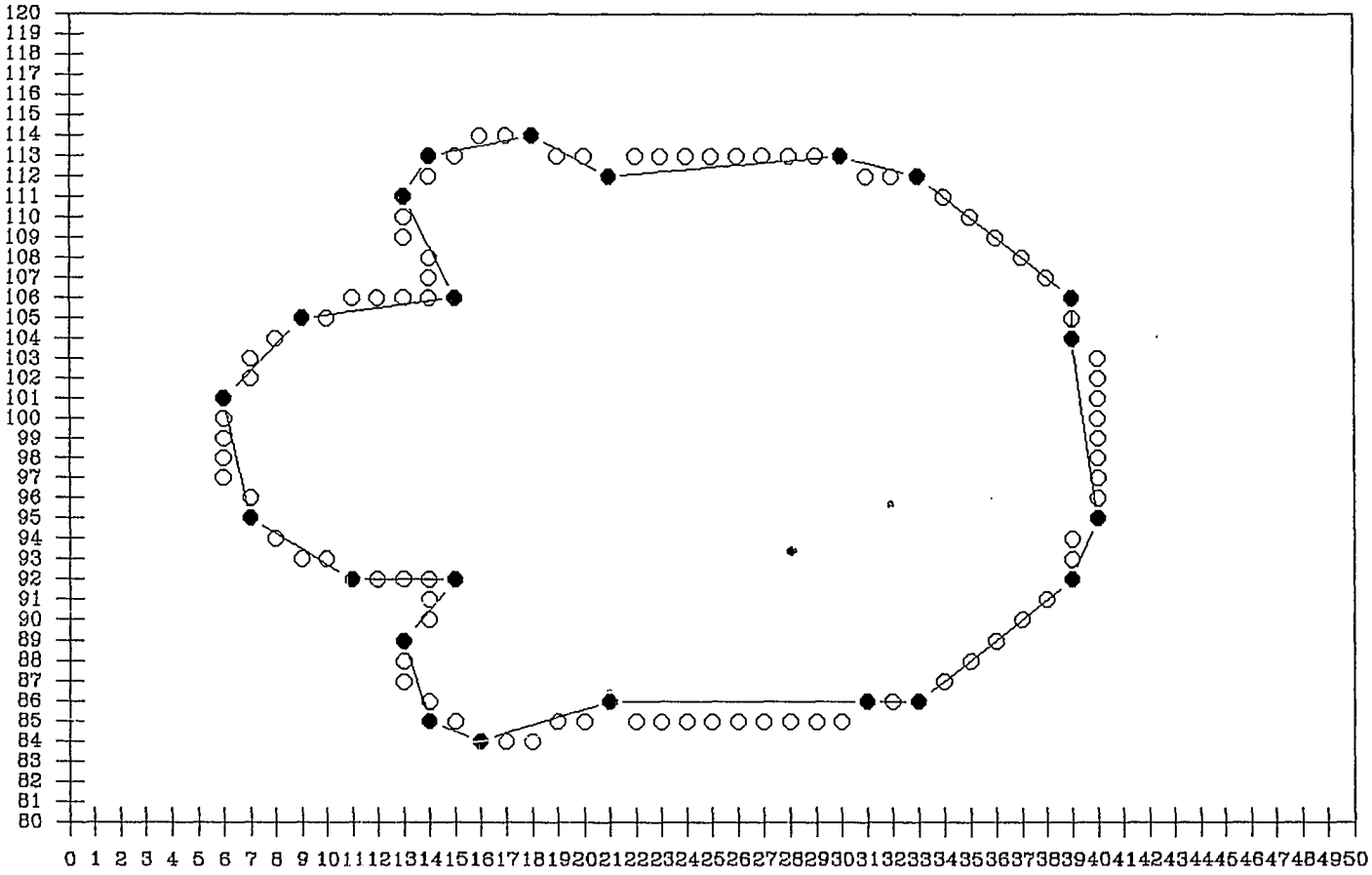


Fig. 5.8 Detected by Teh–Chin Algorithm

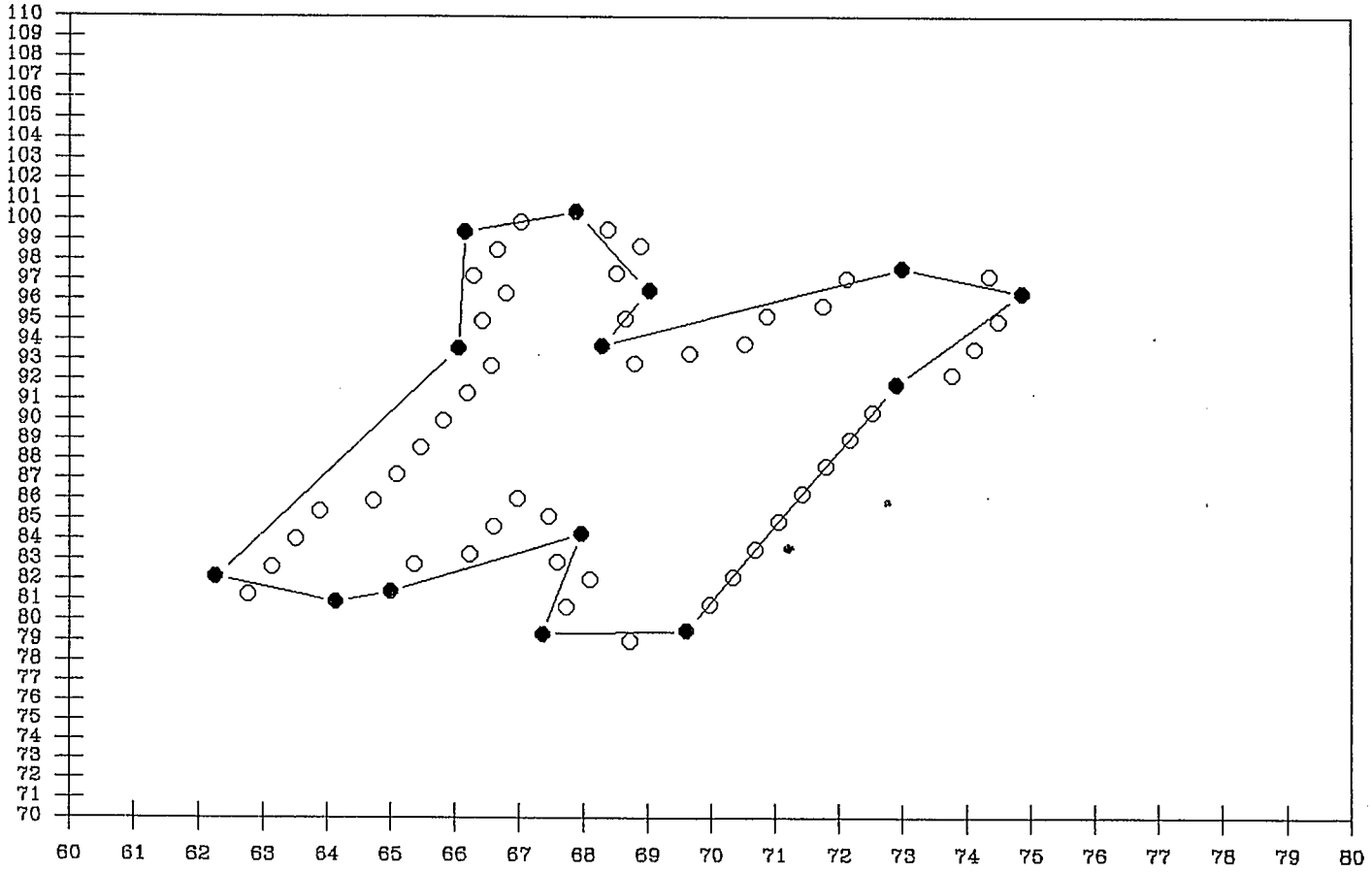


Fig. 5.9 Detected by Teh–Chin Algorithm (Rotation).

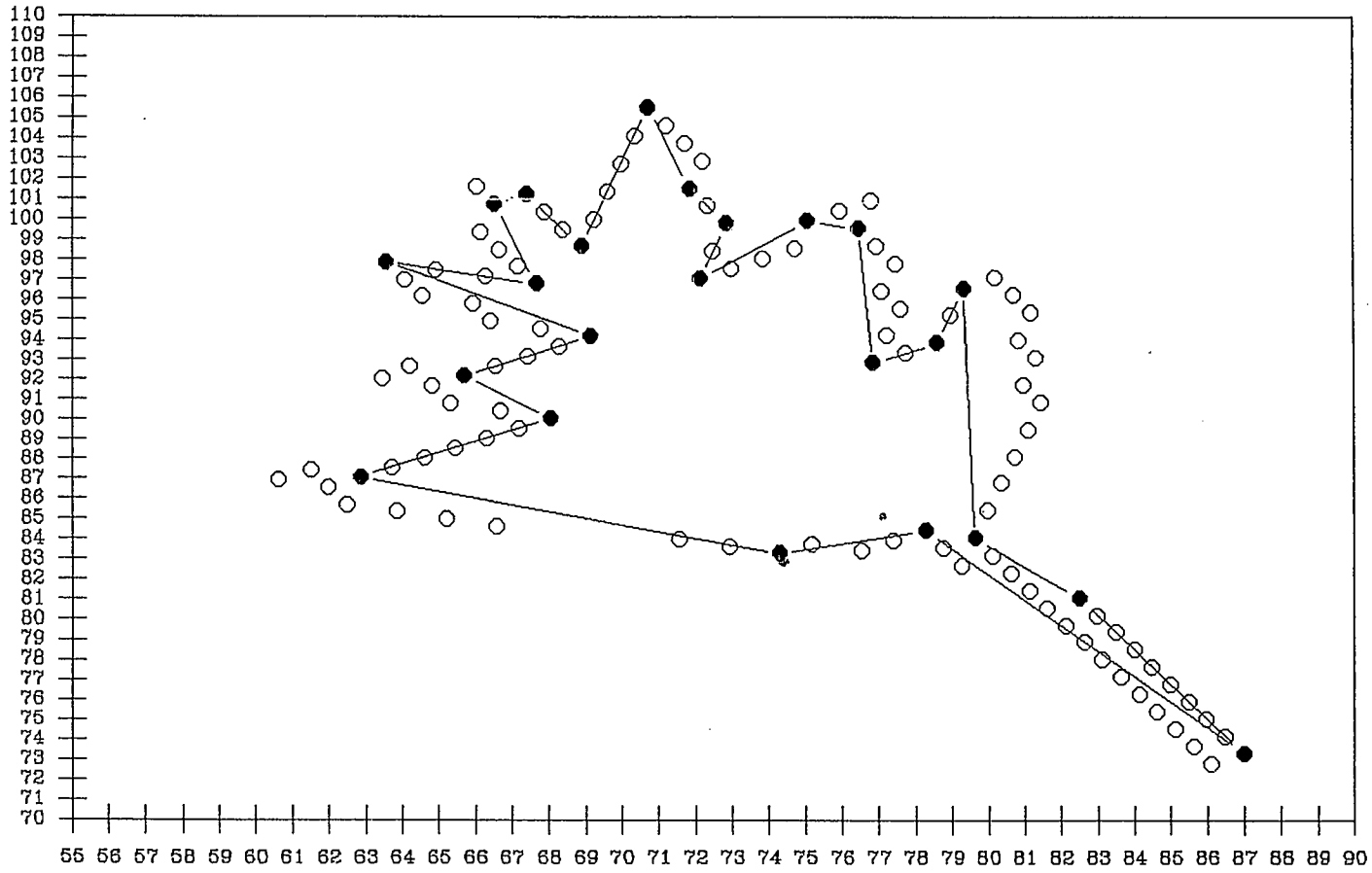


Fig. 5.10 Detected by Teh-Chin Algorithm (Rotation).

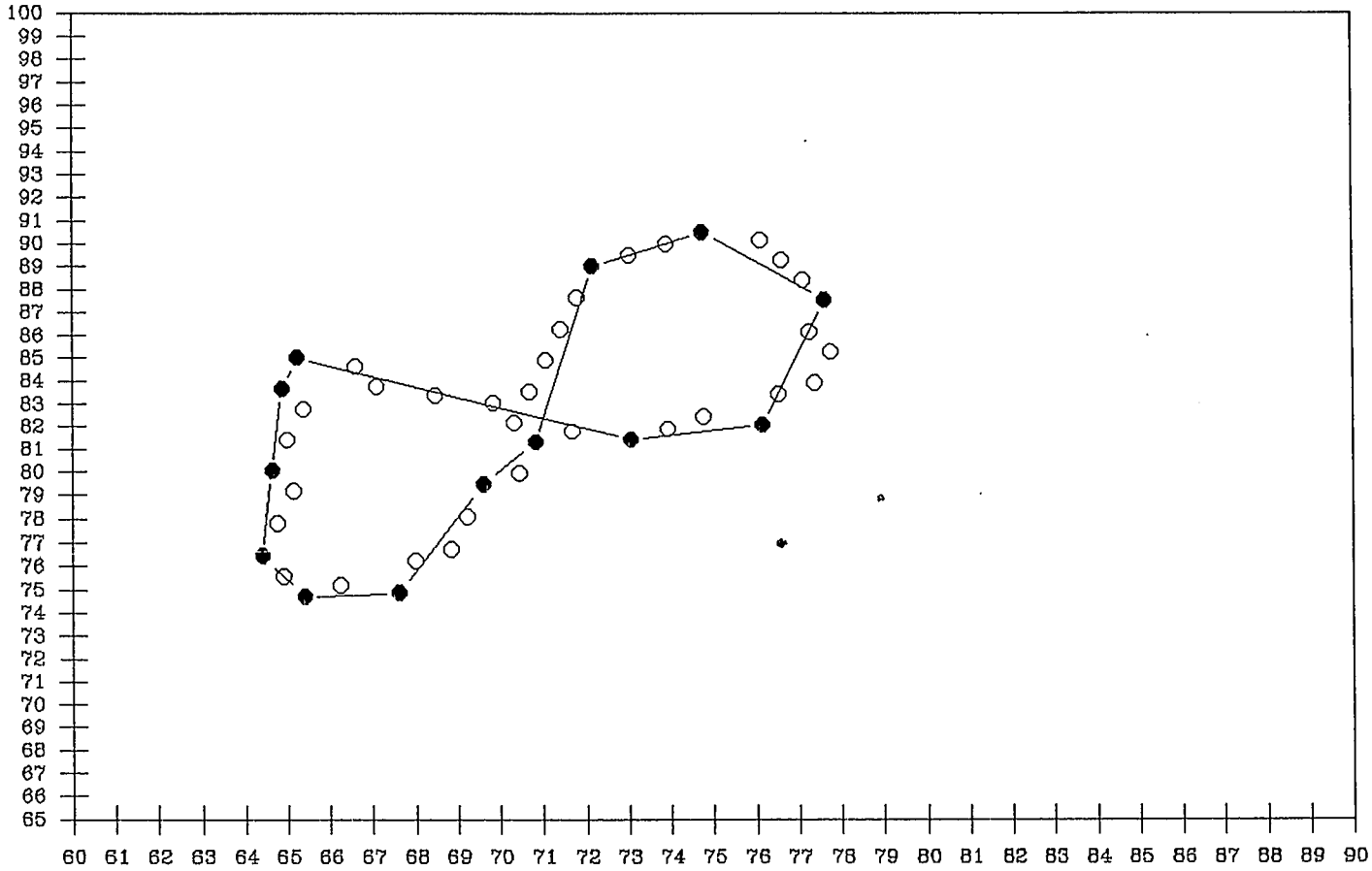


Fig. 5.11 Detected by Teh–Chin Algorithm (Rotation).

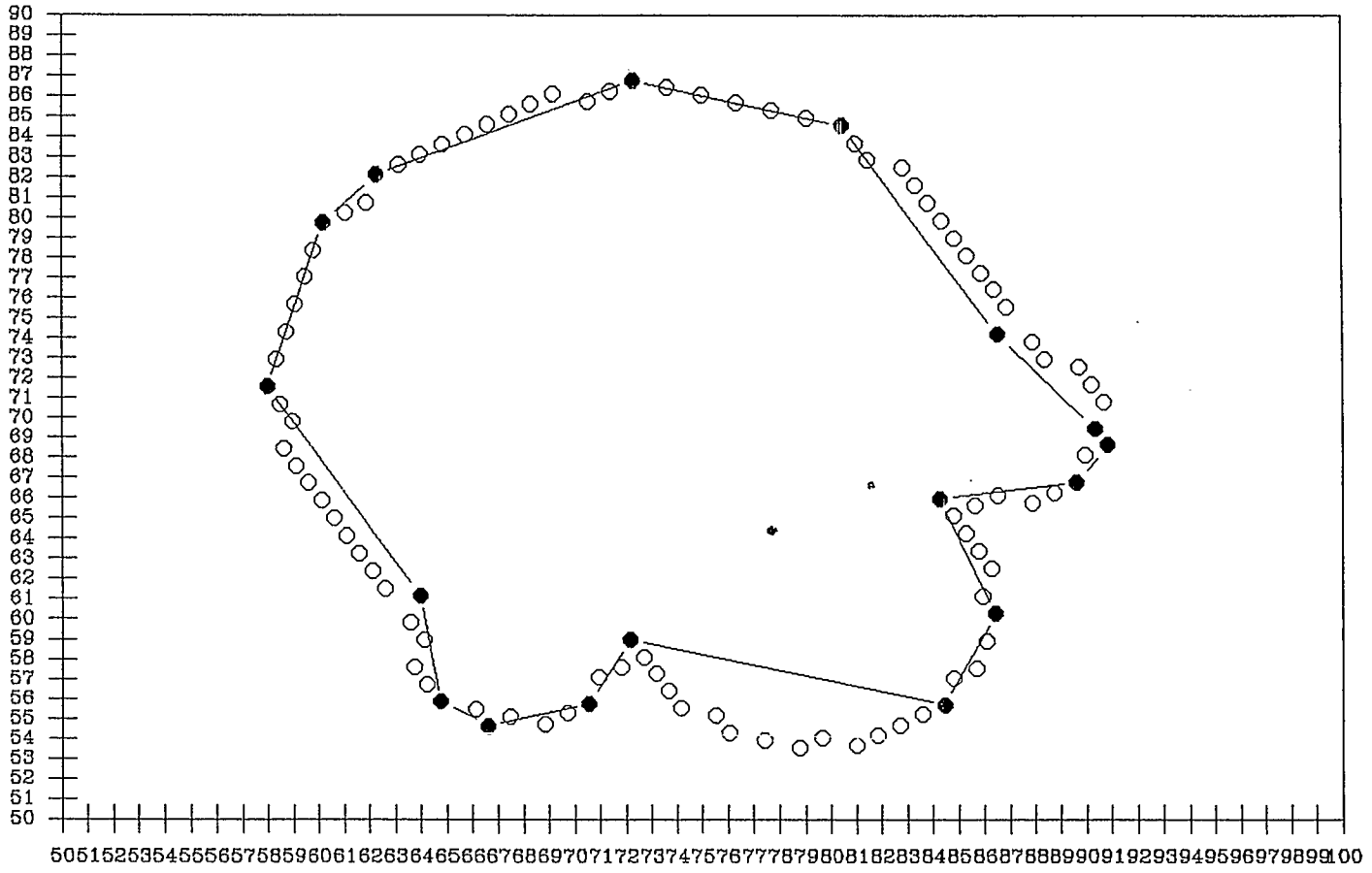


Fig. 5.12 Detected by Teh–Chin Algorithm (Rotation).

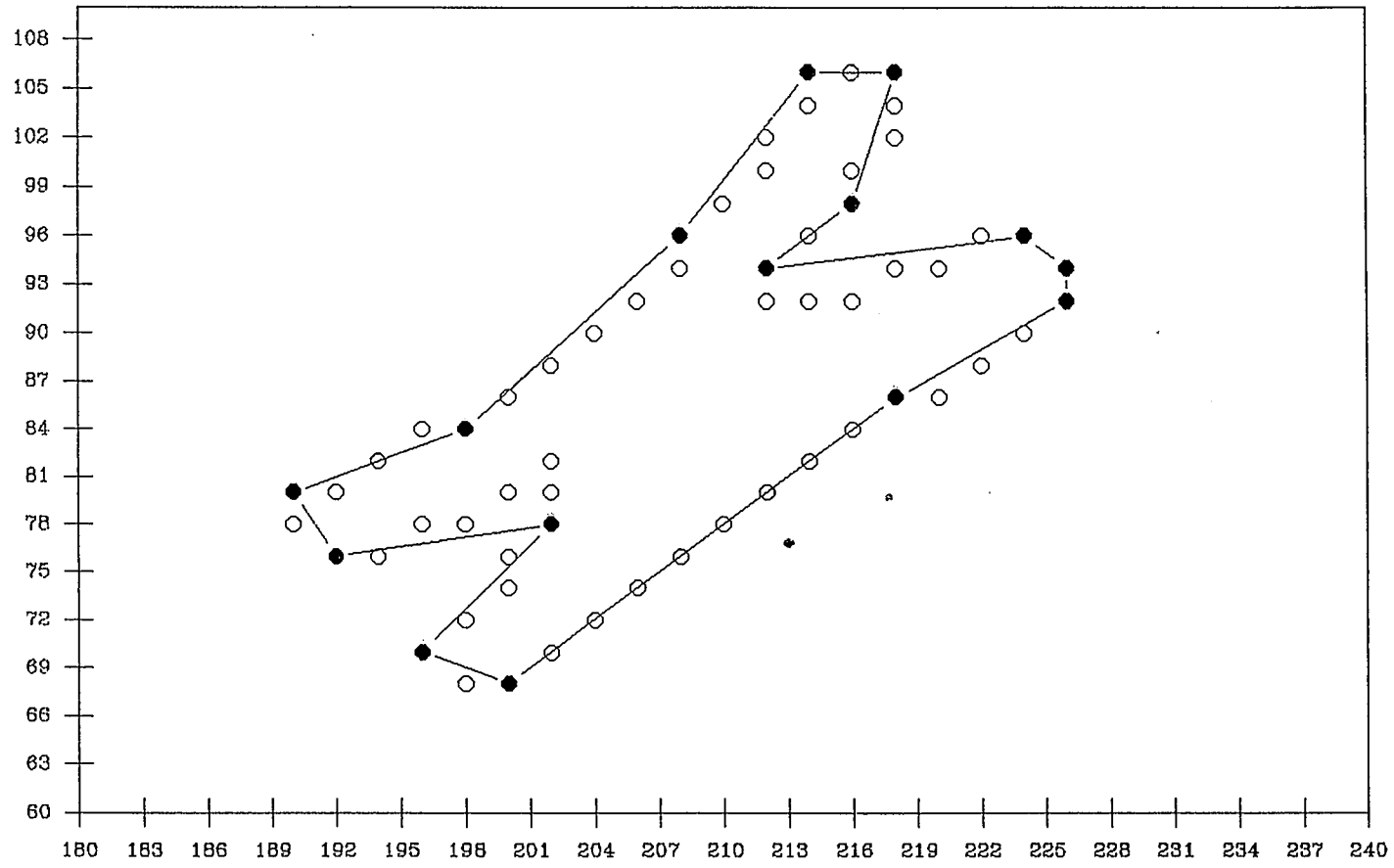


Fig. 5.13 Detected by Teh–Chin Algorithm (Scaling).

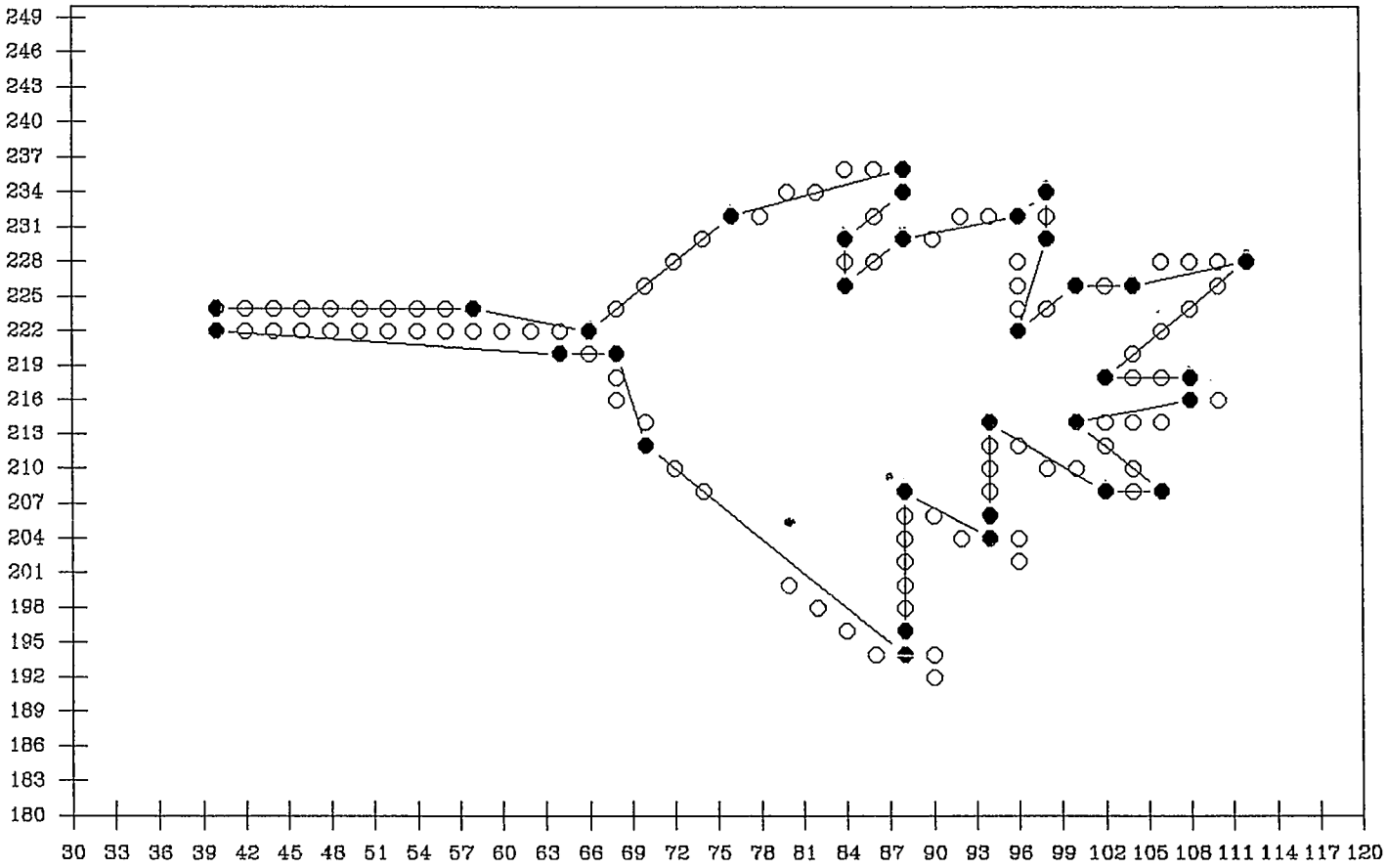


Fig. 5.14 Detected by Teh–Chin Algorithm (Scaling).

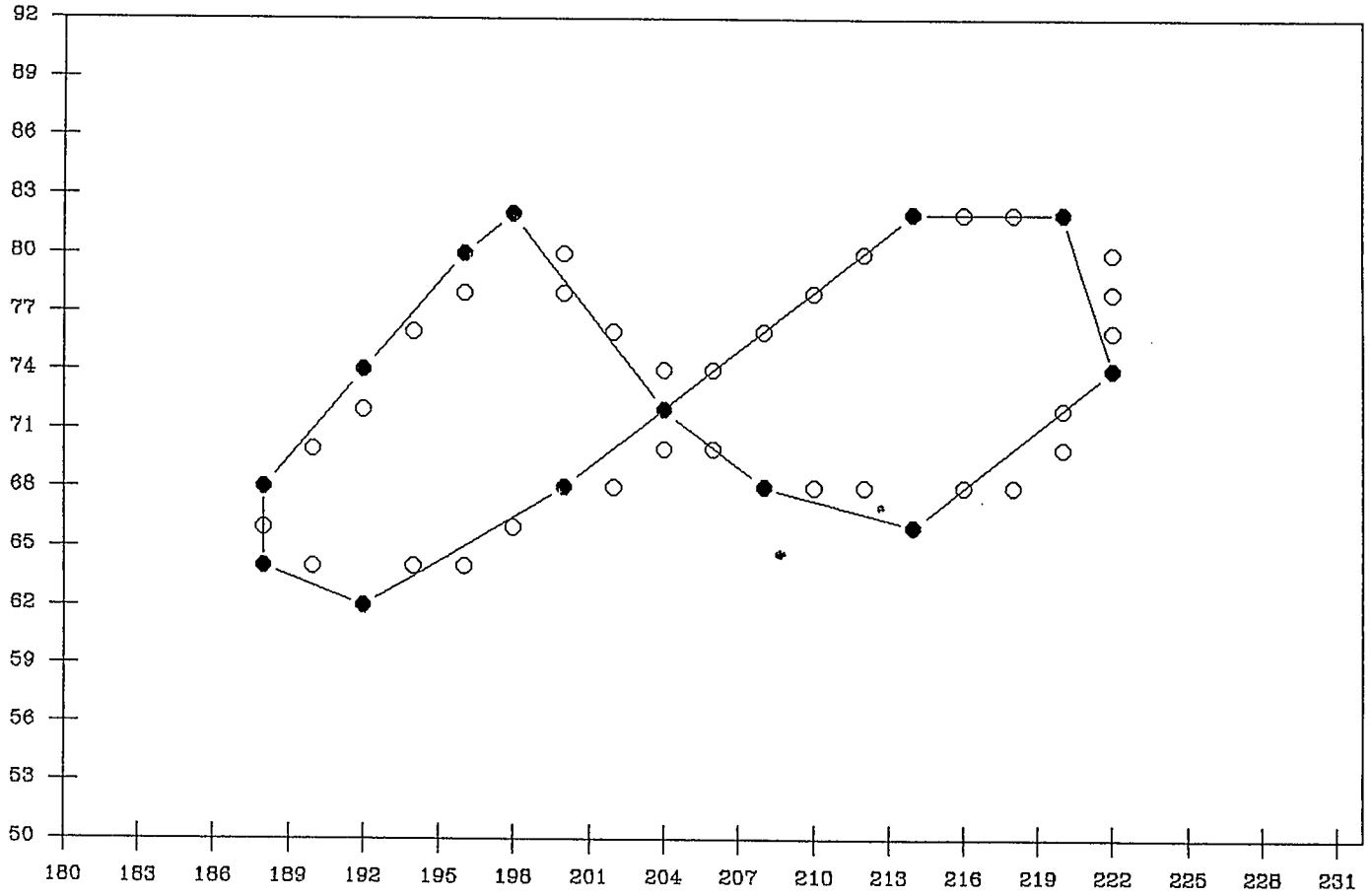


Fig. 5.15 Detected by Teh–Chin Algorithm (Scaling).

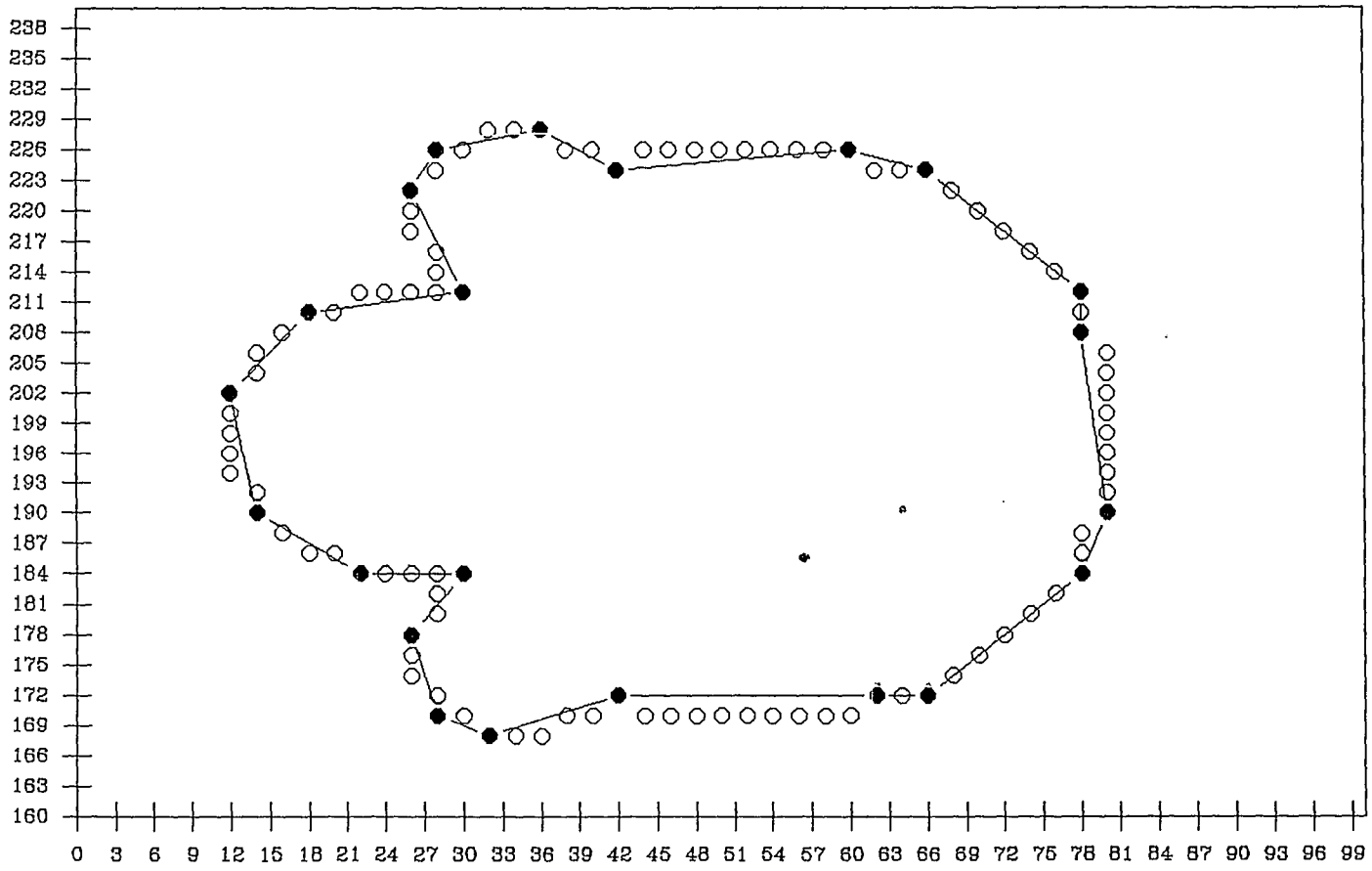


Fig. 5.16 Detected by Teh-Chin Algorithm (Scaling).

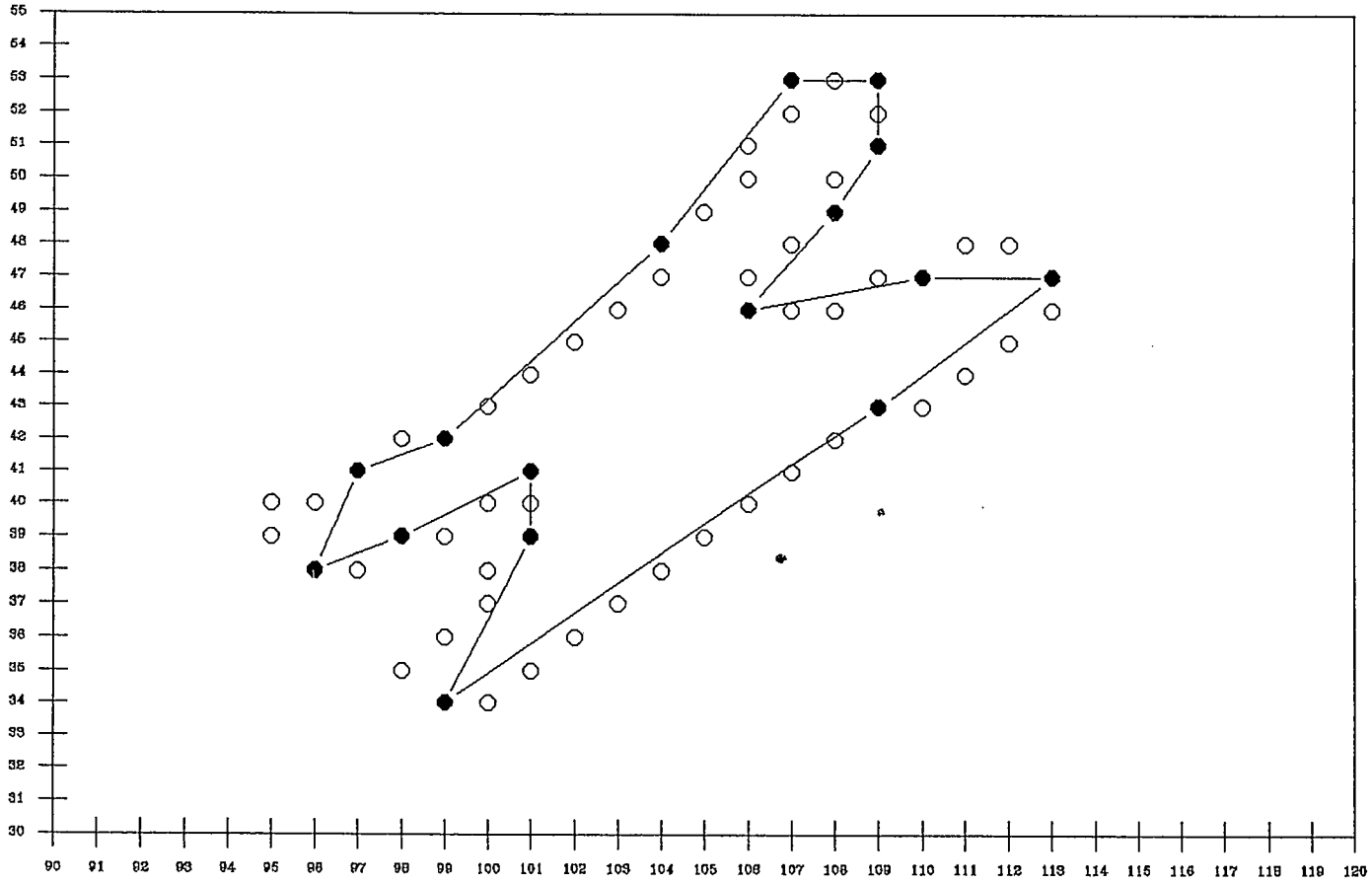


Fig. 5.17 Detected by First Method.

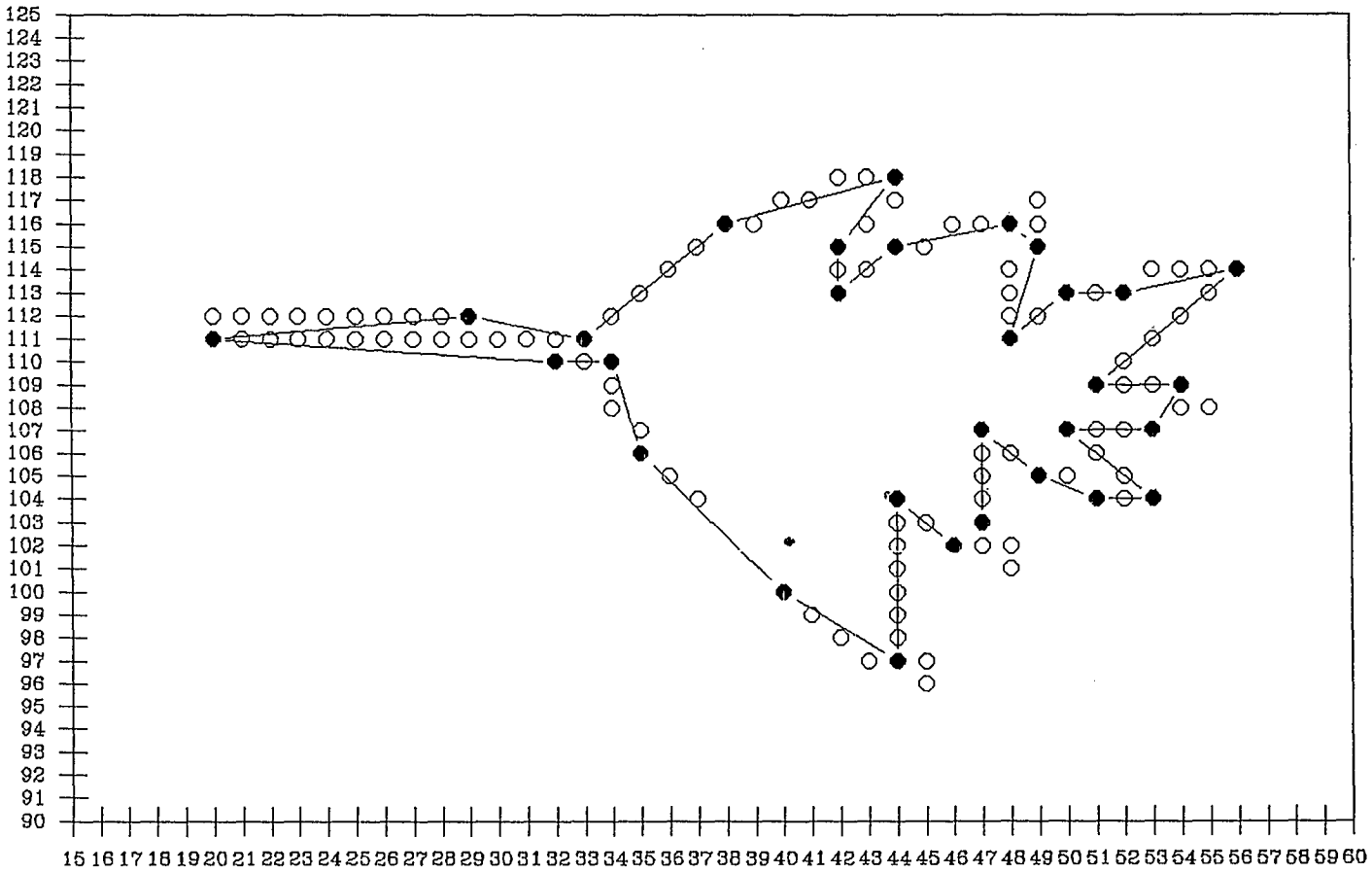


Fig. 5.18 Detected by First Method.

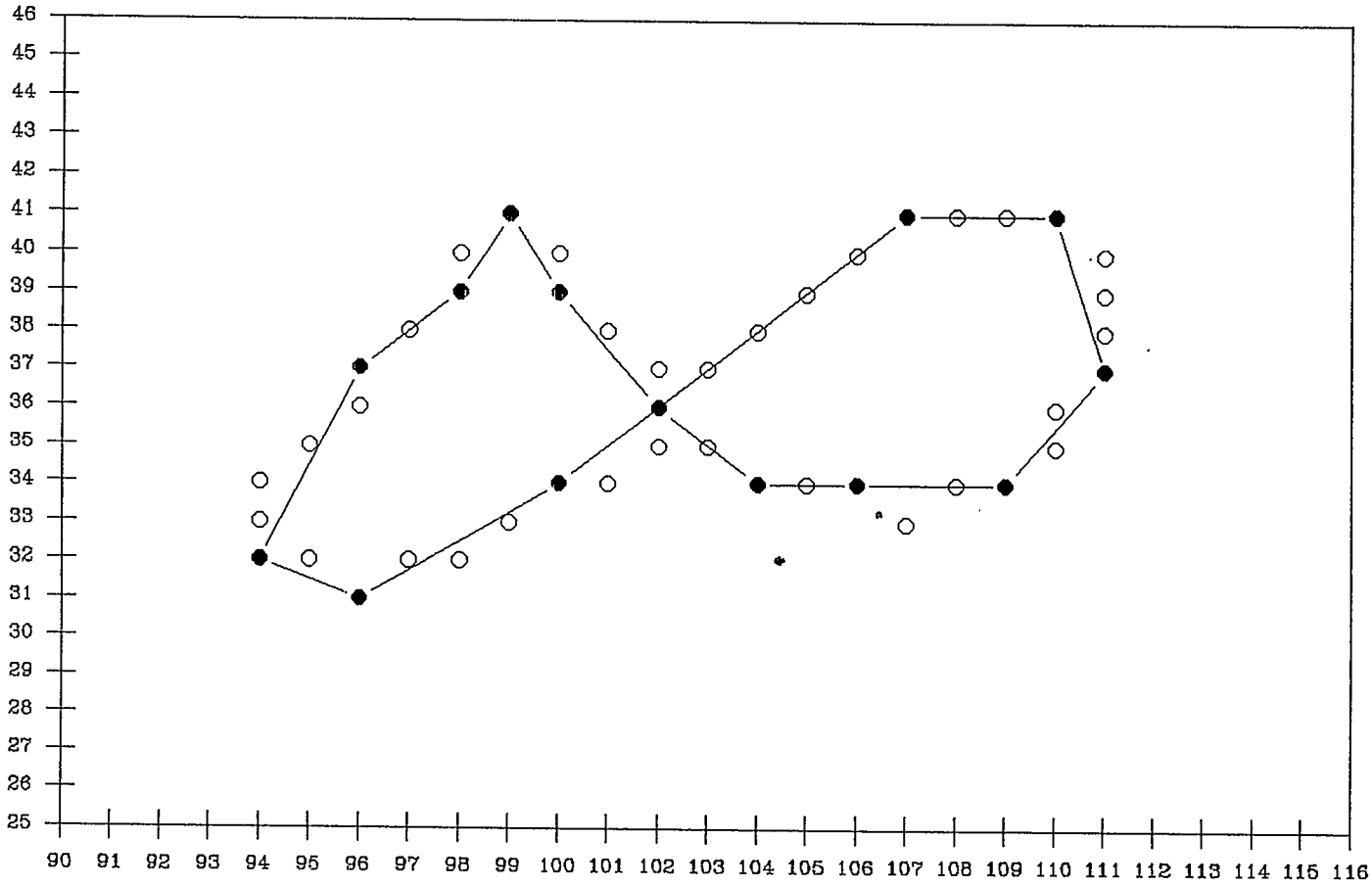


Fig. 5.19 Detected by First Method.

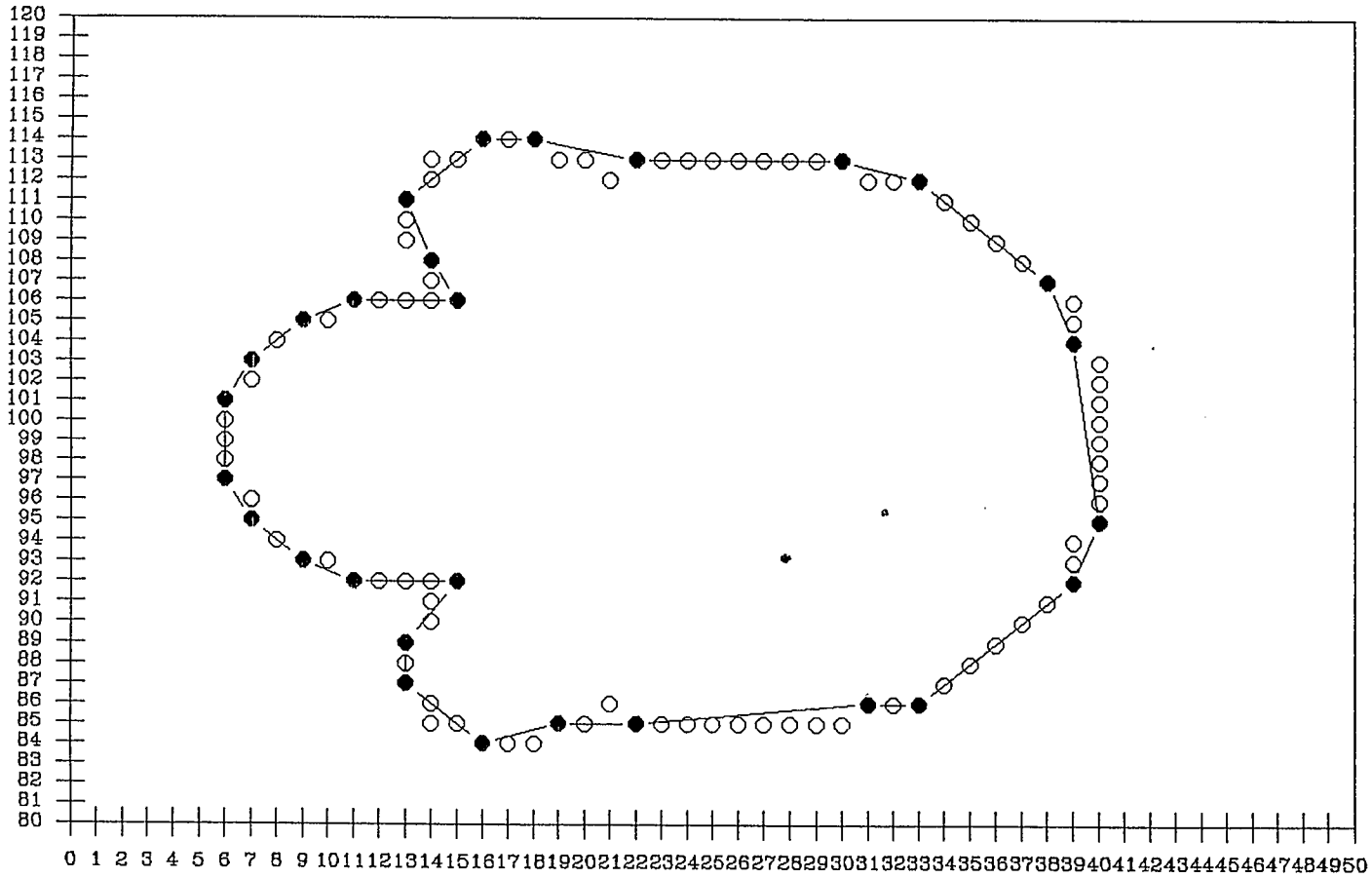


Fig. 5.20 Detected by First Method.

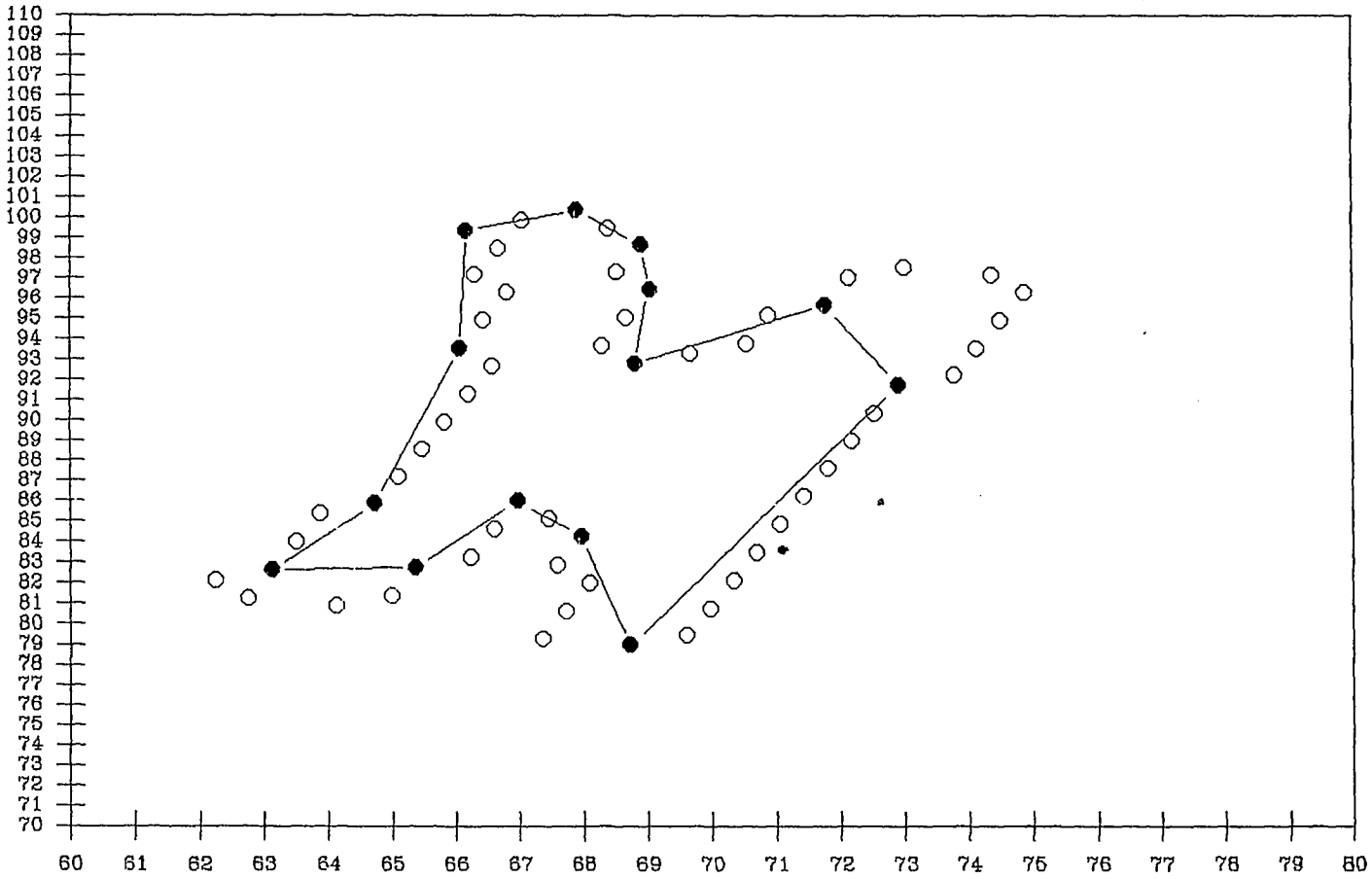


Fig. 5.21 Detected by First Method (Rotation).

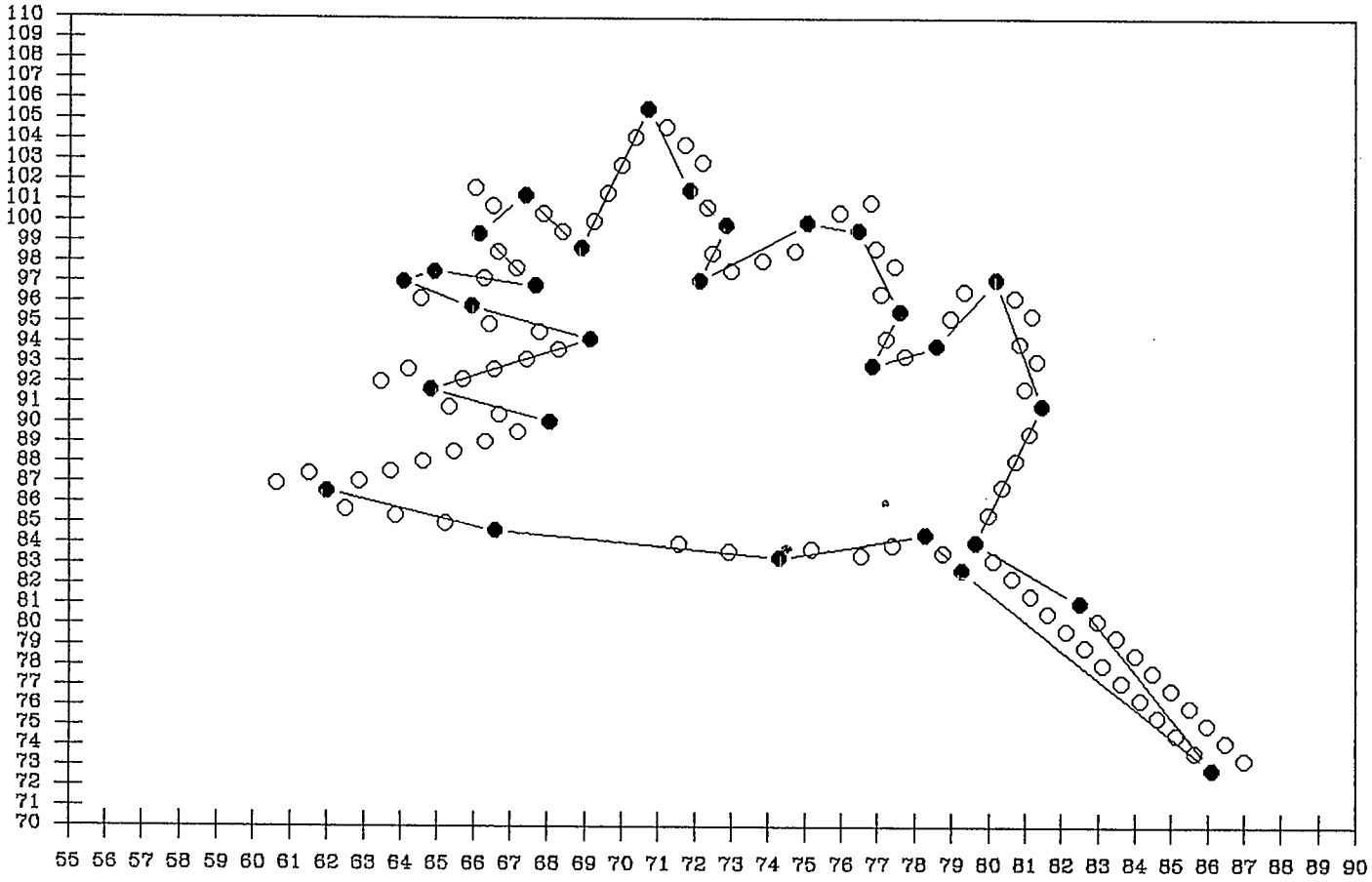


Fig. 5.22 Detected by First Method (Rotation).

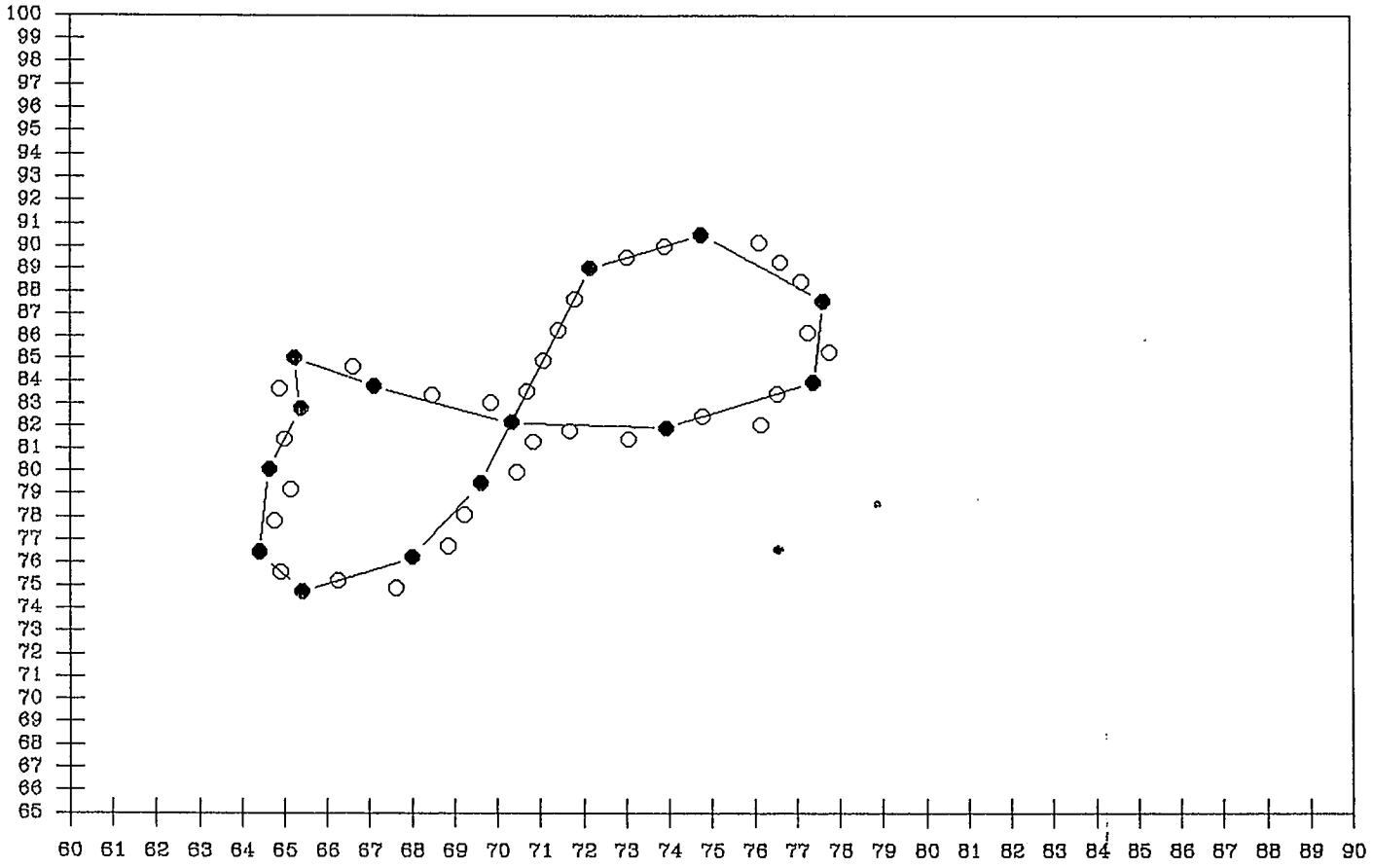


Fig. 5.23 Detected by First Method (Rotation).

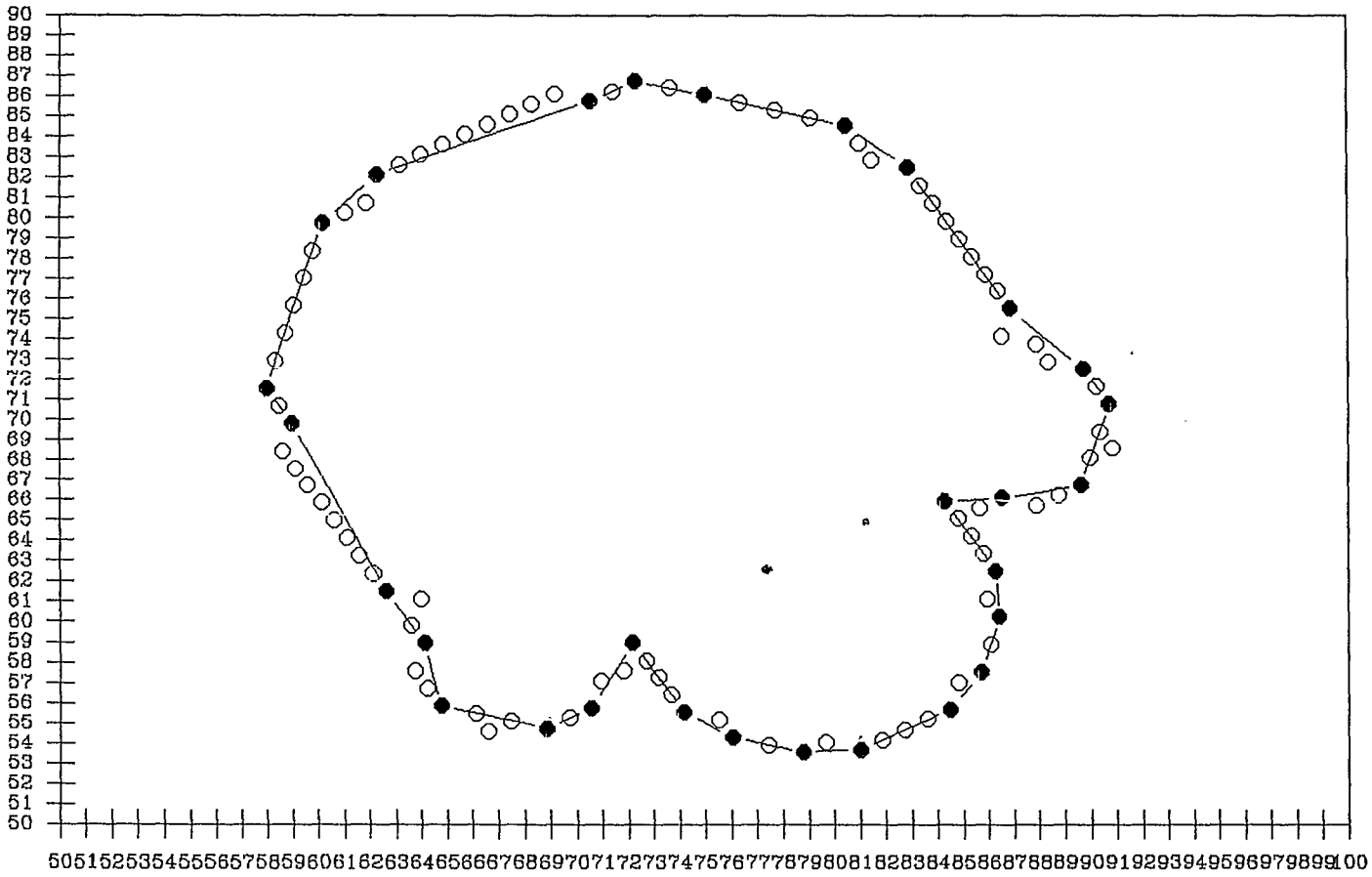


Fig. 5.24 Detected by First Method (Rotation).

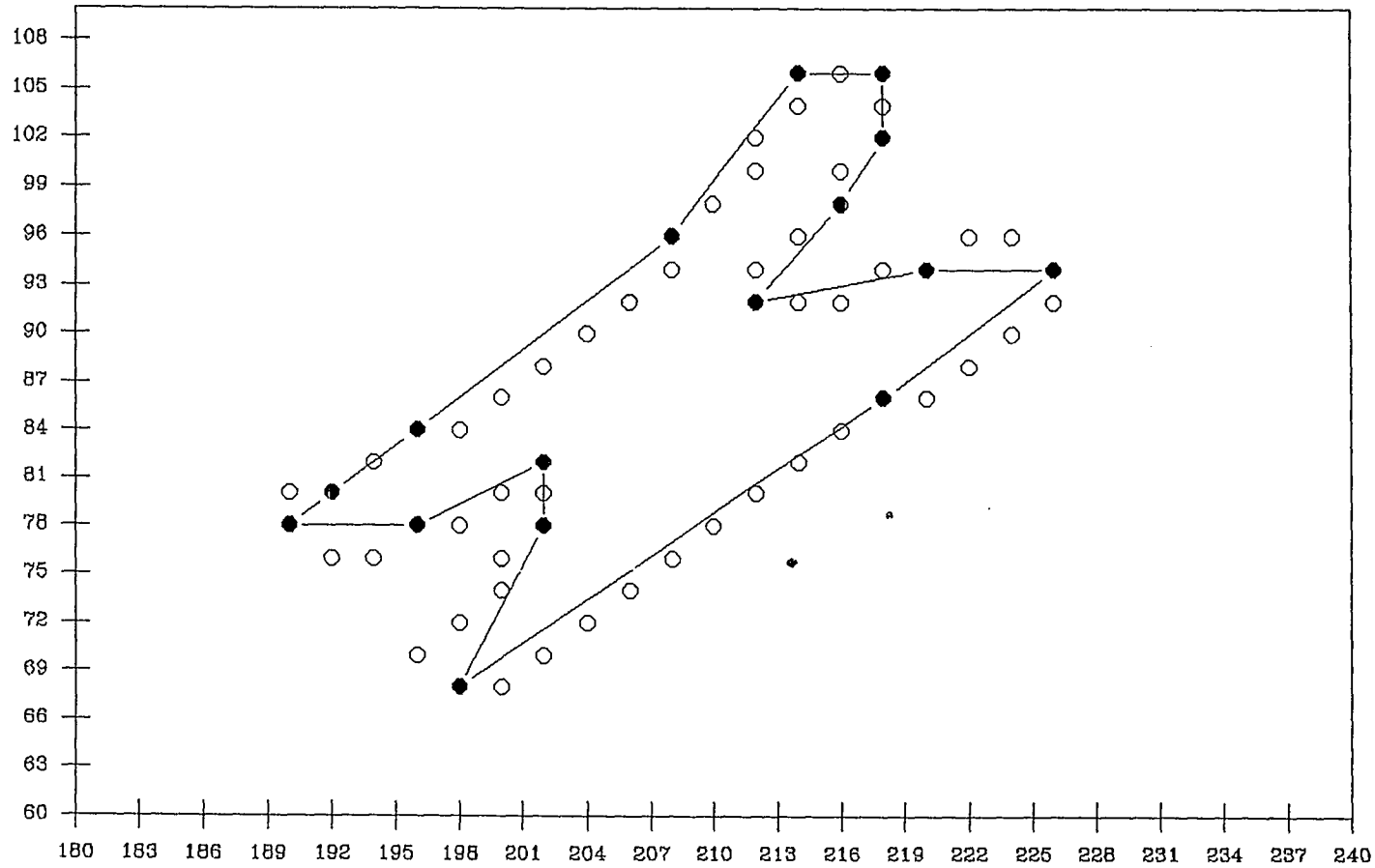


Fig. 5.25 Detected by First Method (Scaling).

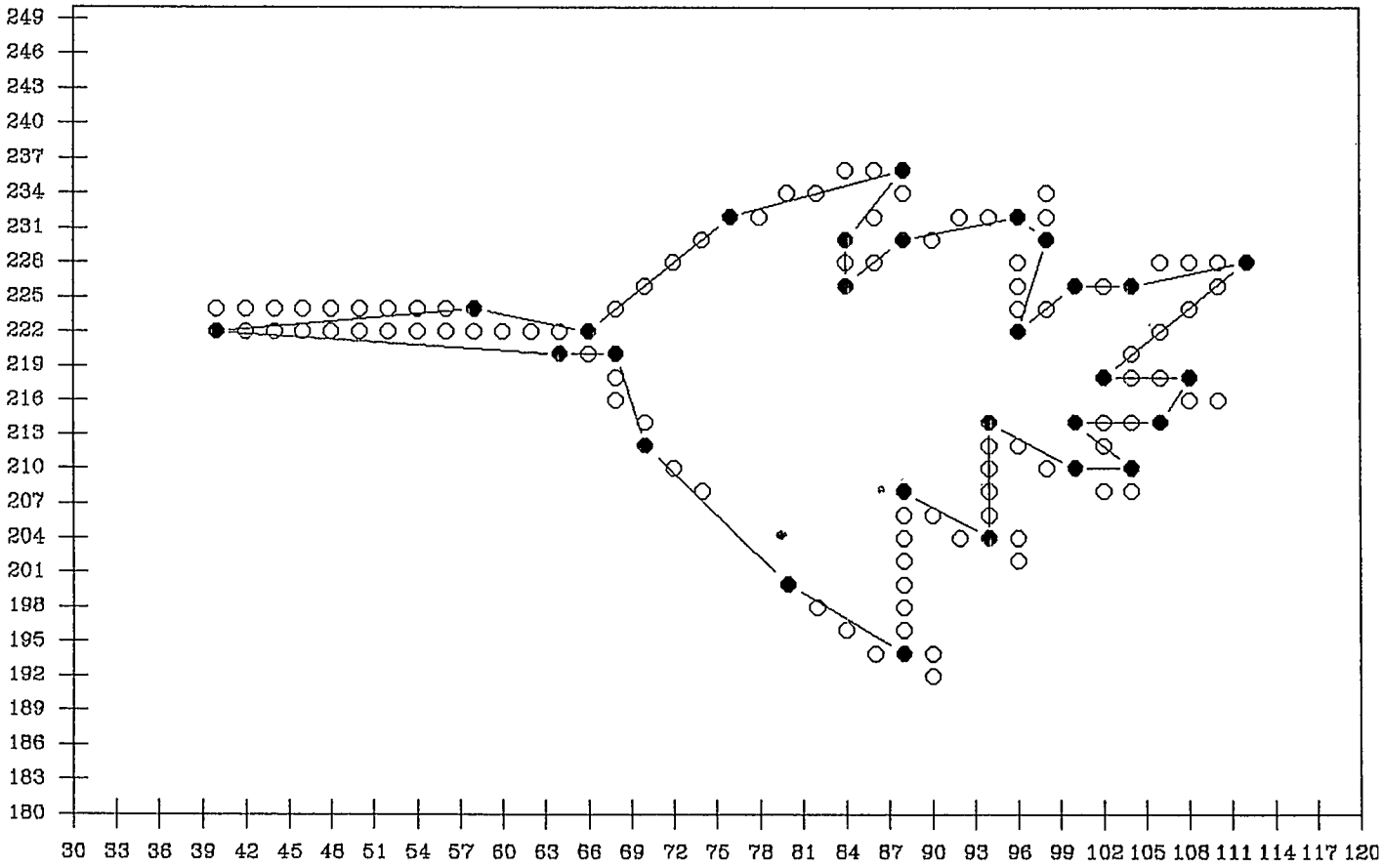


Fig. 5.26 Detected by First Method (Scaling).

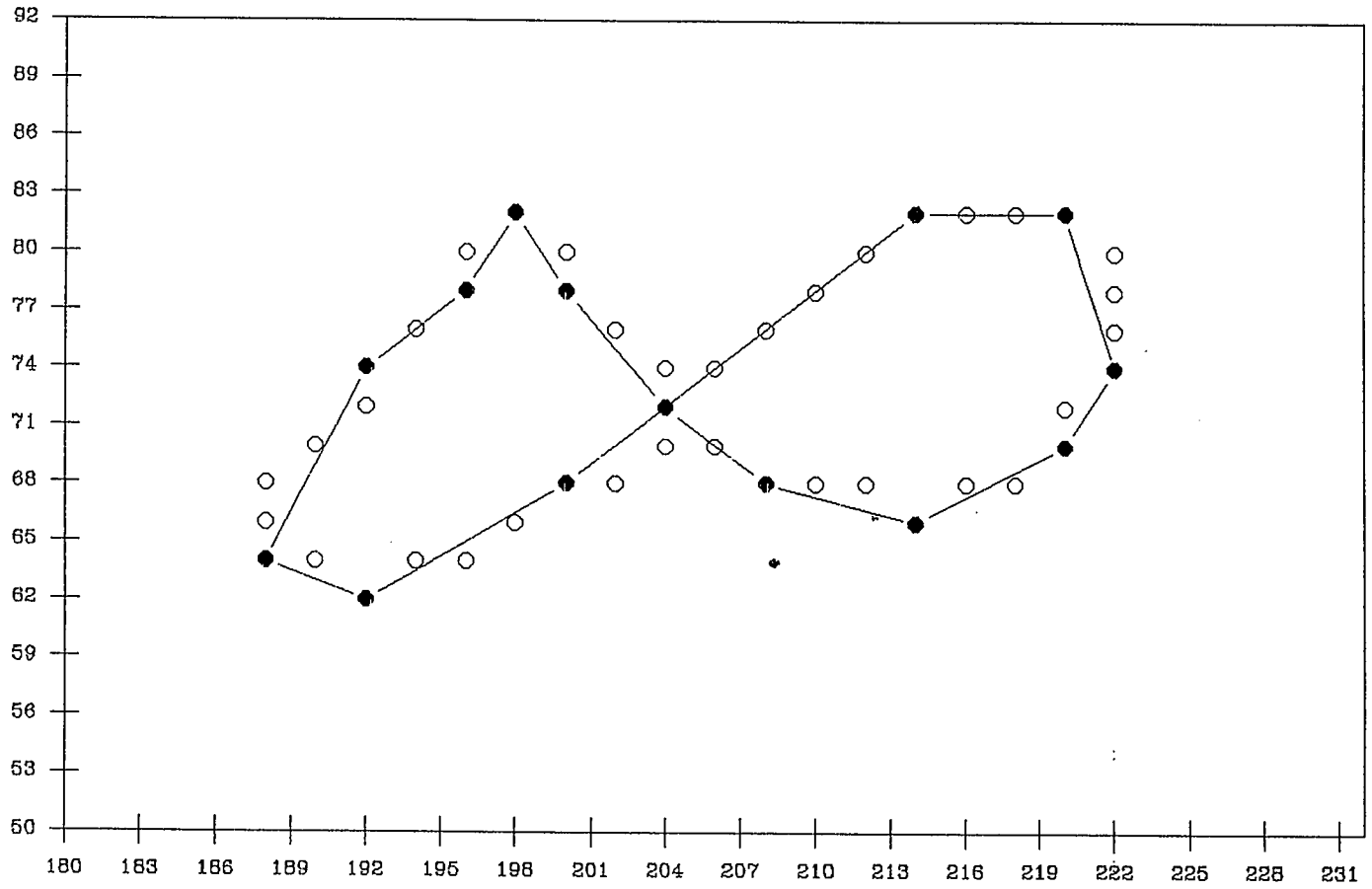


Fig. 5.27 Detected by First Method (Scaling).

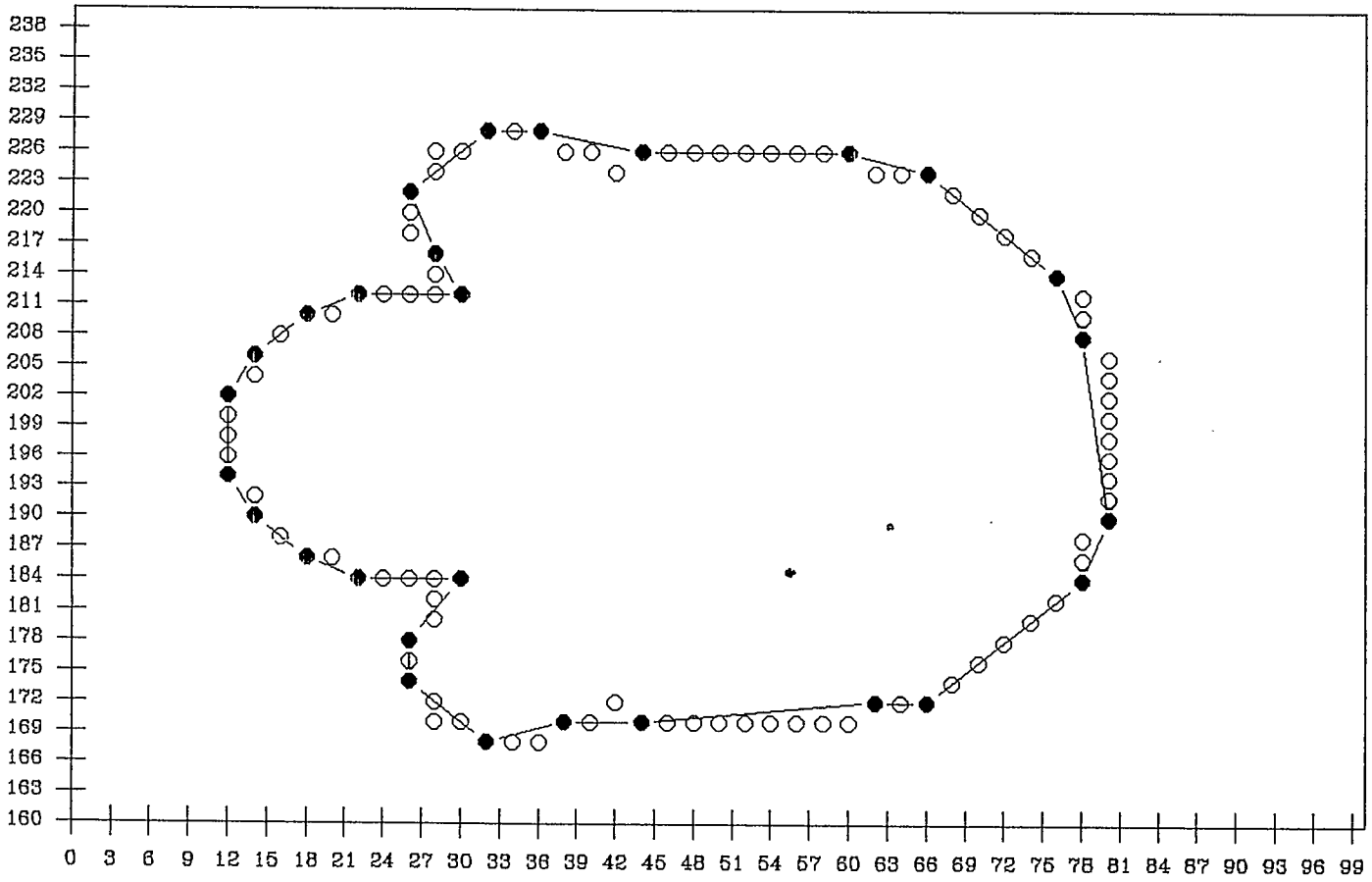


Fig. 5.28 Detected by First Method (Scaling).

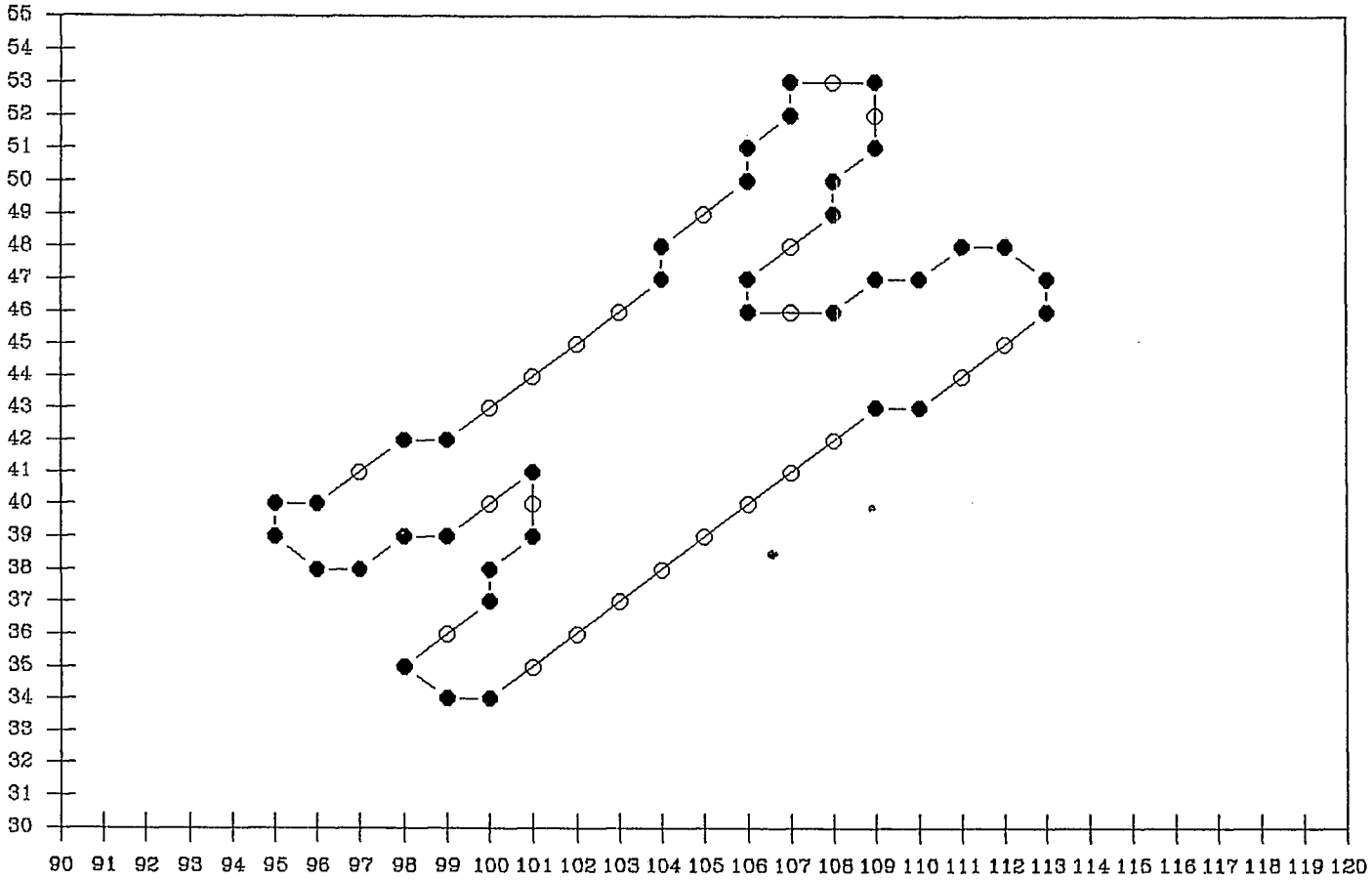


Fig. 5.29 Detected by Second Method .

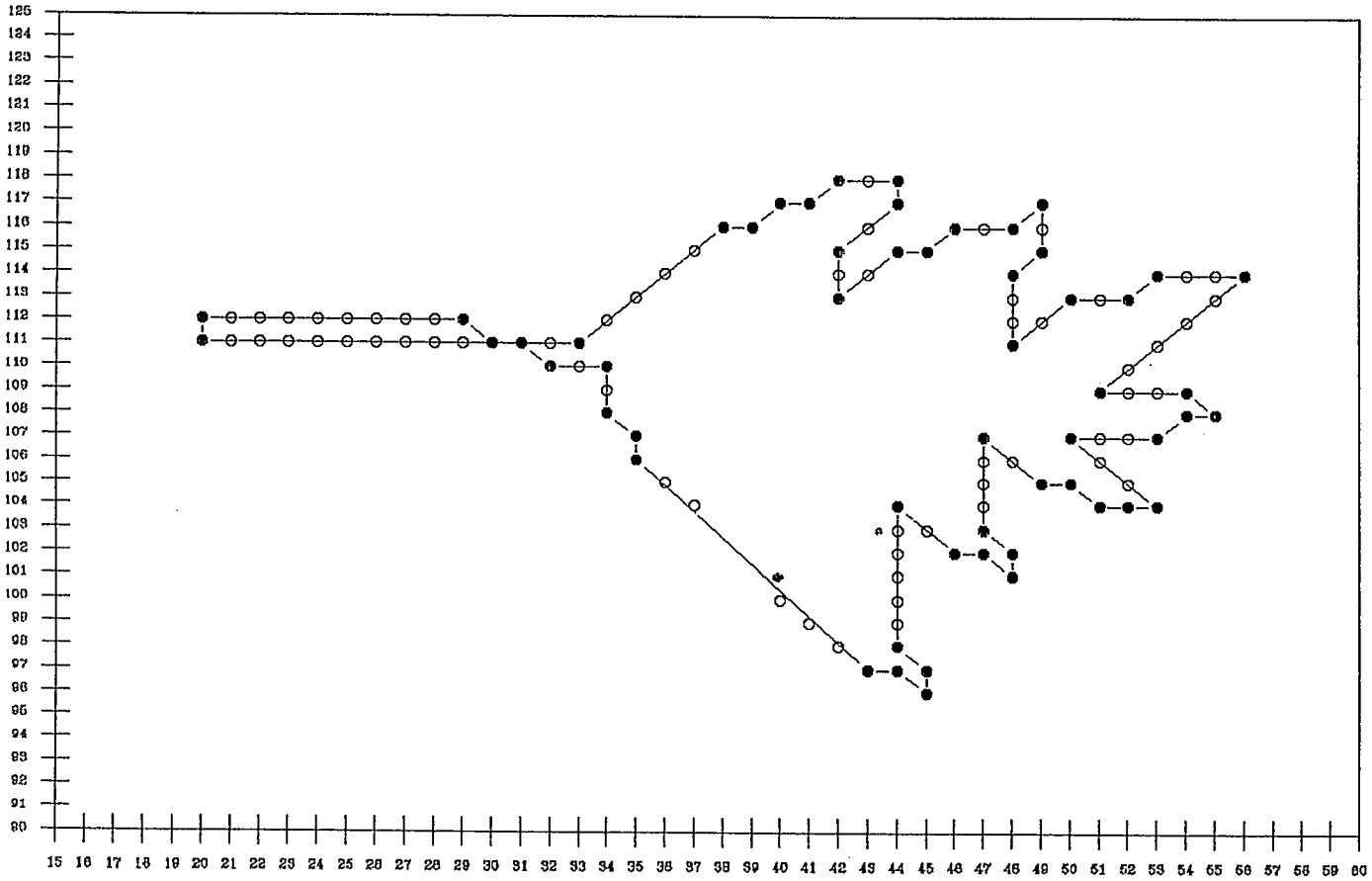


Fig. 5.30 Detected by Second Method.

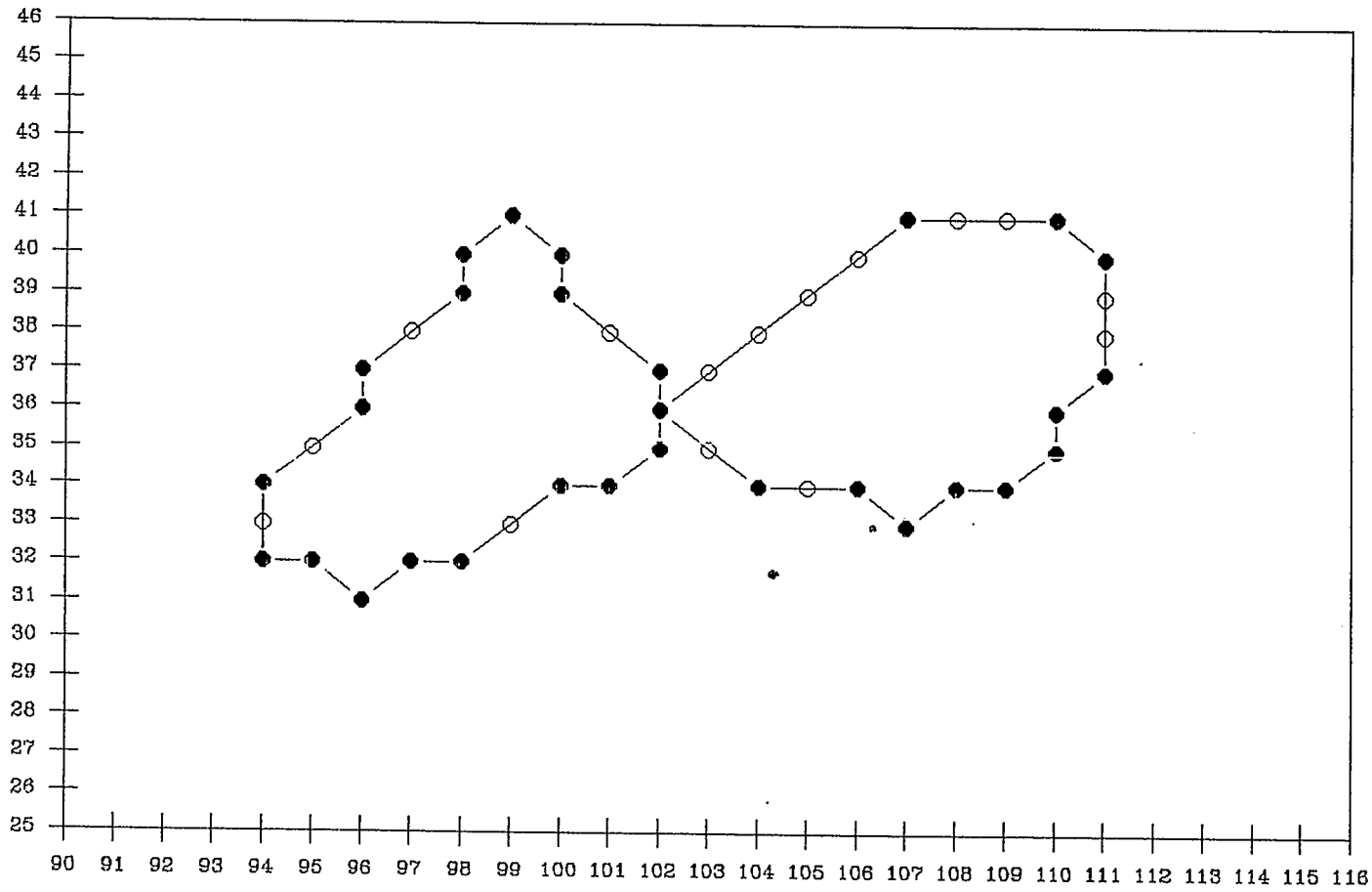


Fig. 5.31 Detected by Second Method.

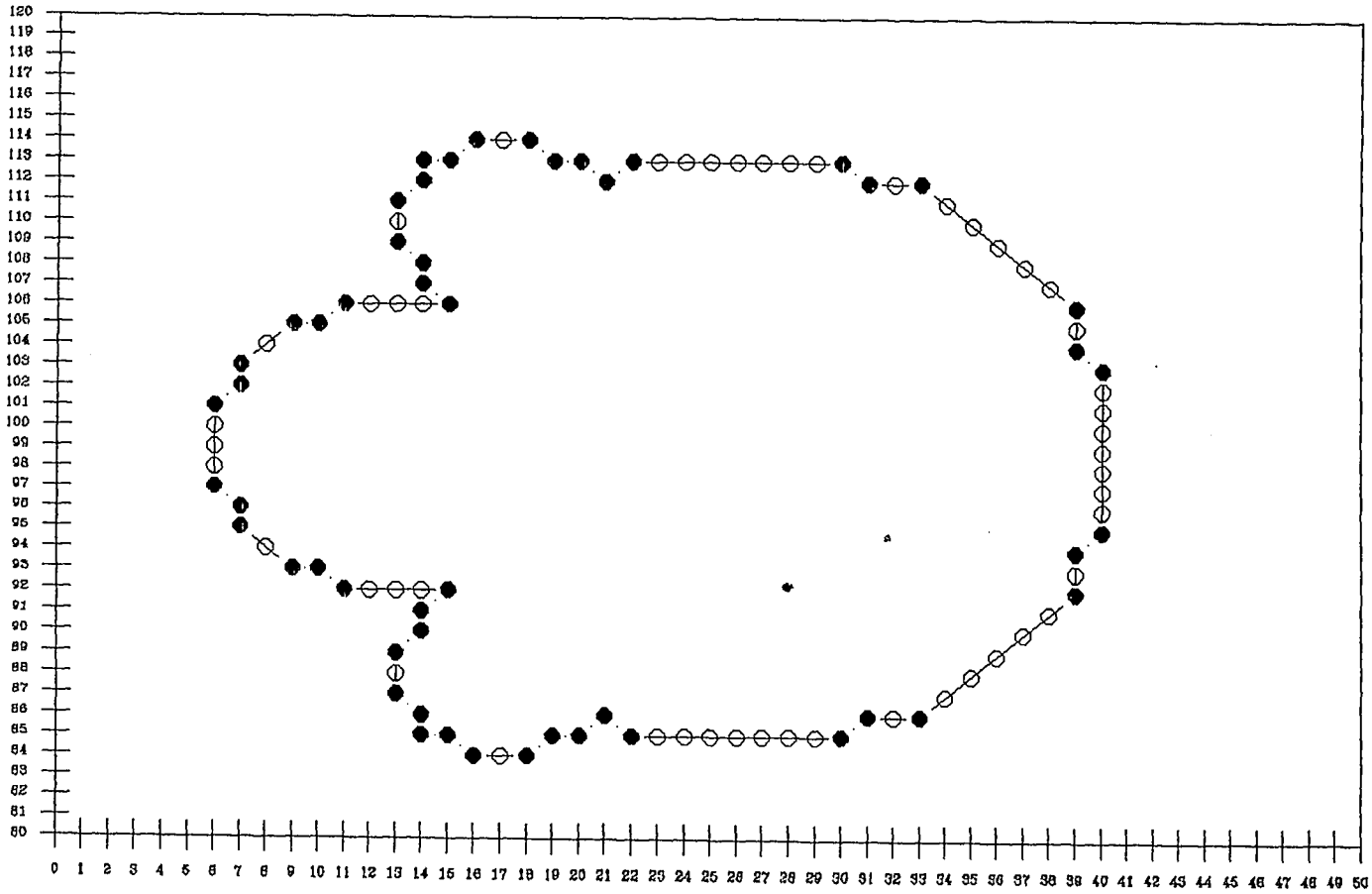


Fig. 5.32 Detected by Second Method.

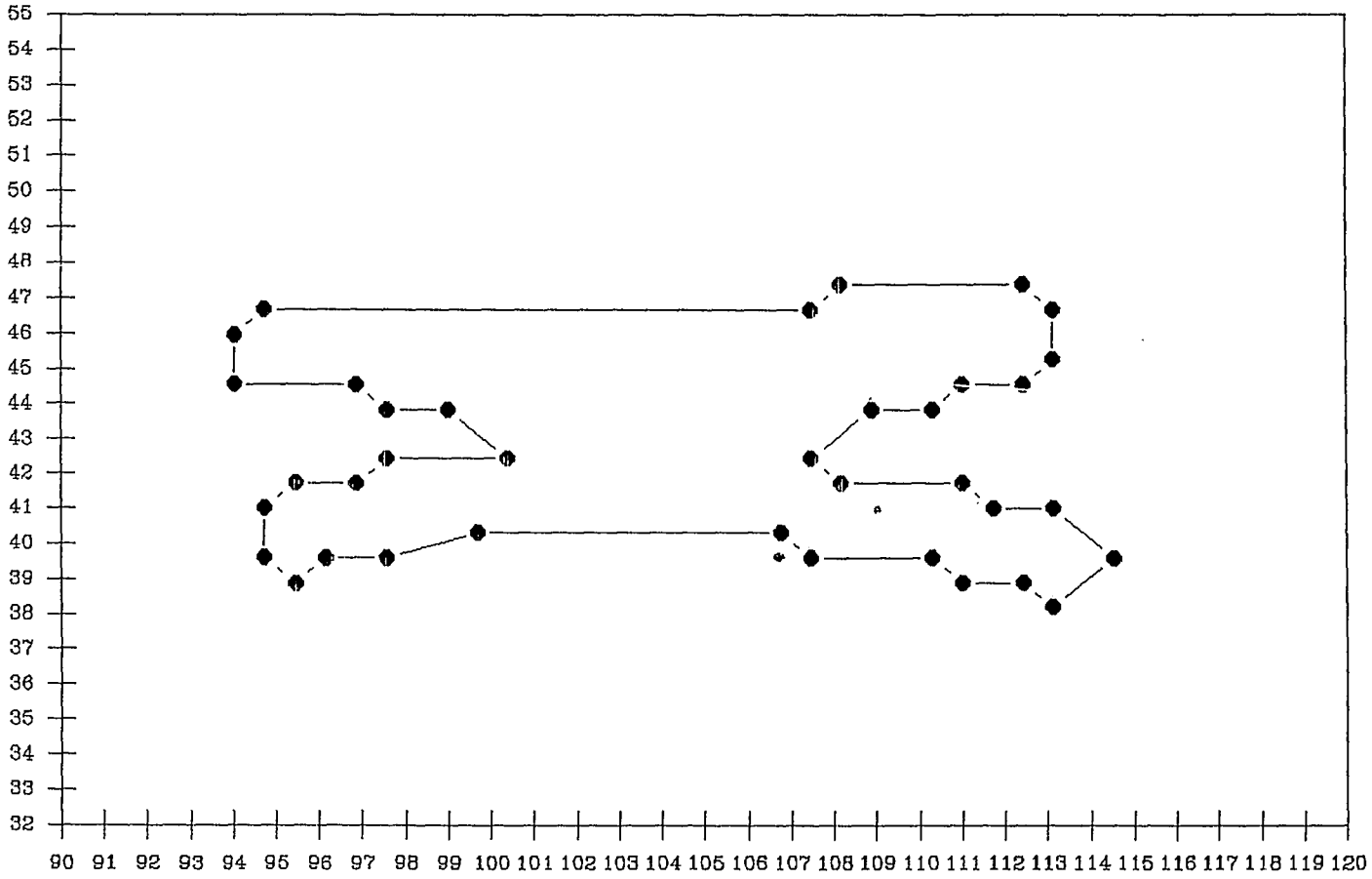


Fig. 5.33 Detected by Second Method (Rotation).

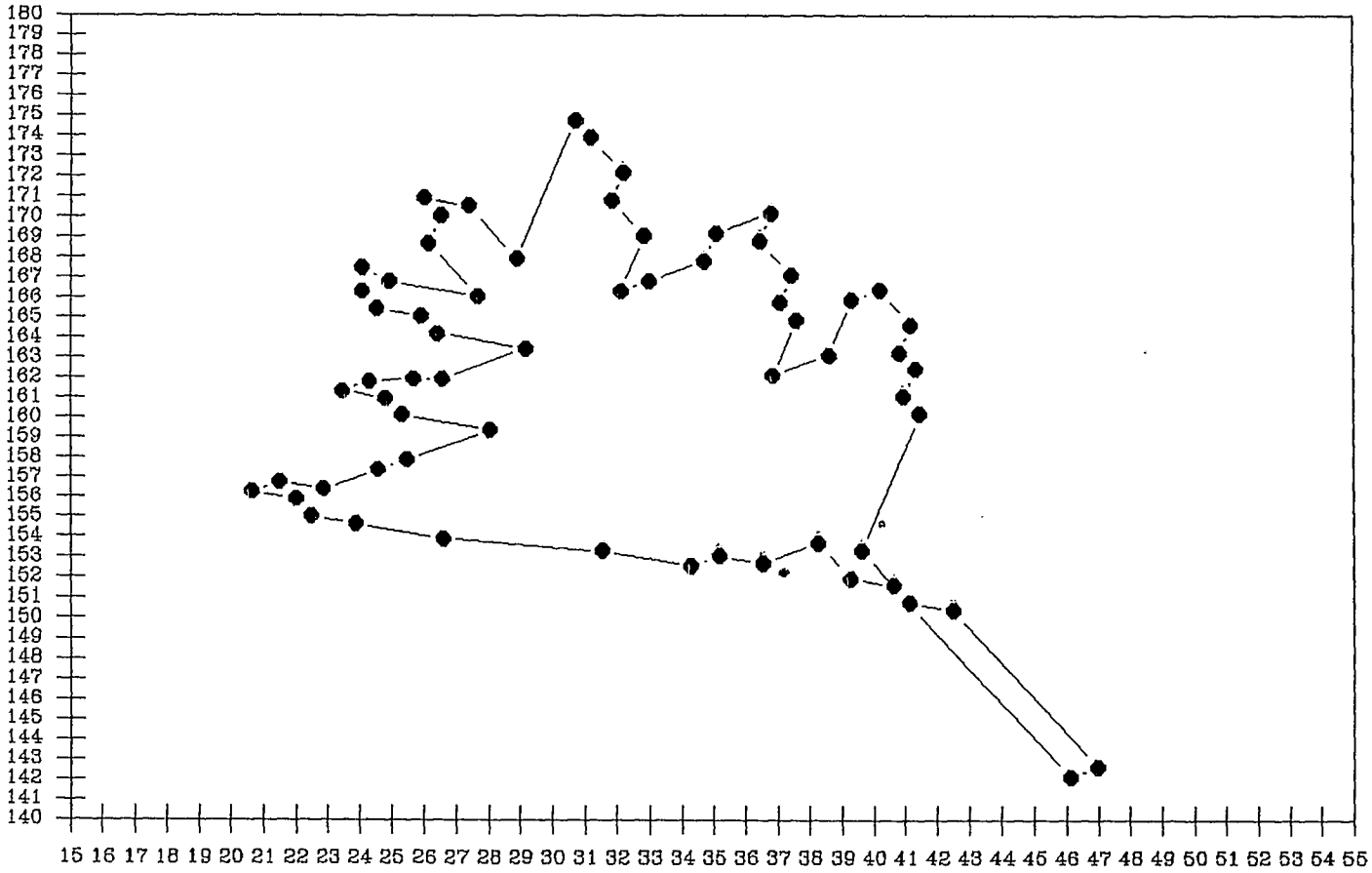


Fig. 5.34 Detected by Second Method (Rotation).

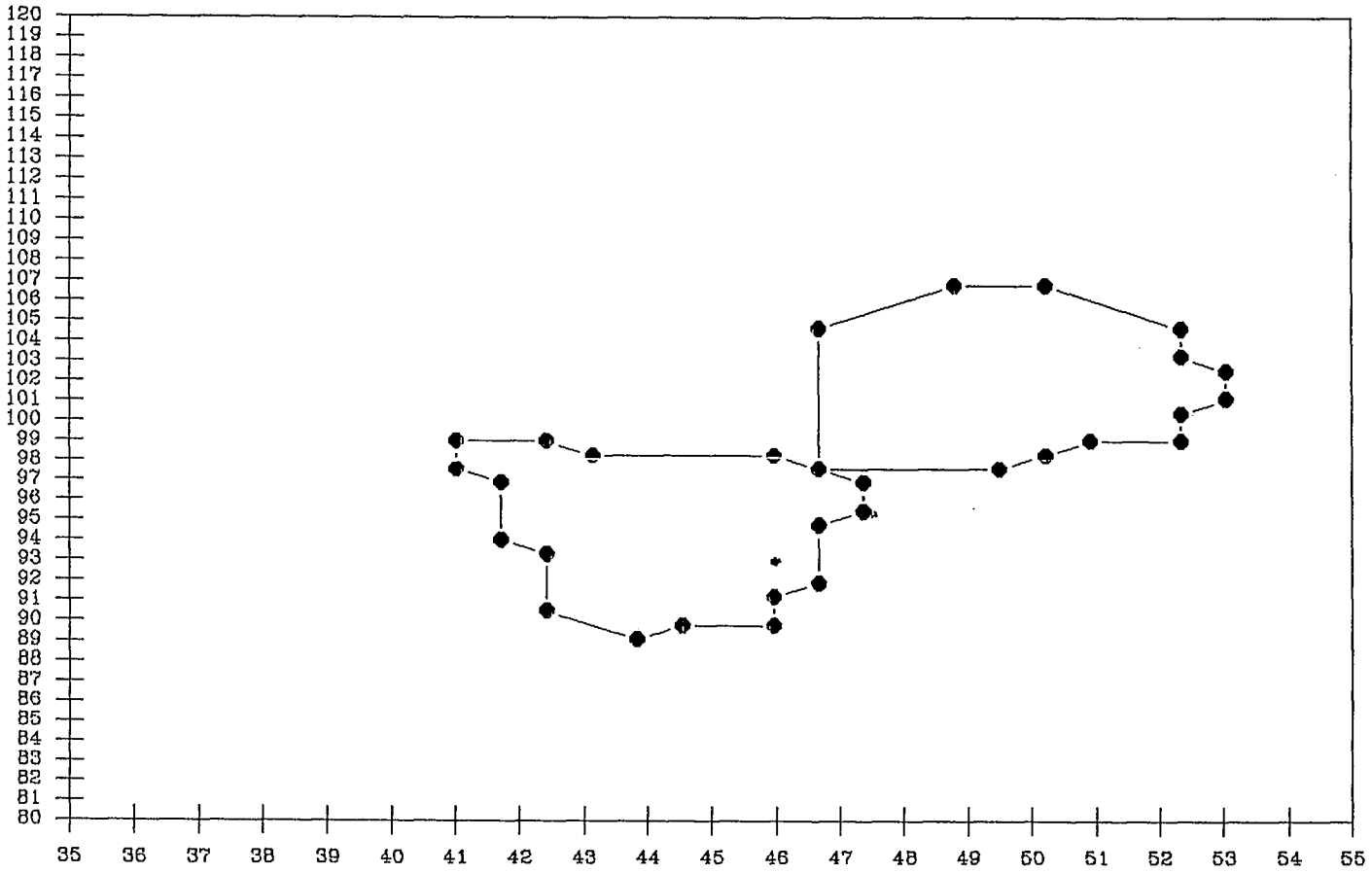


Fig. 5.35 Detected by Second Method (Rotation).

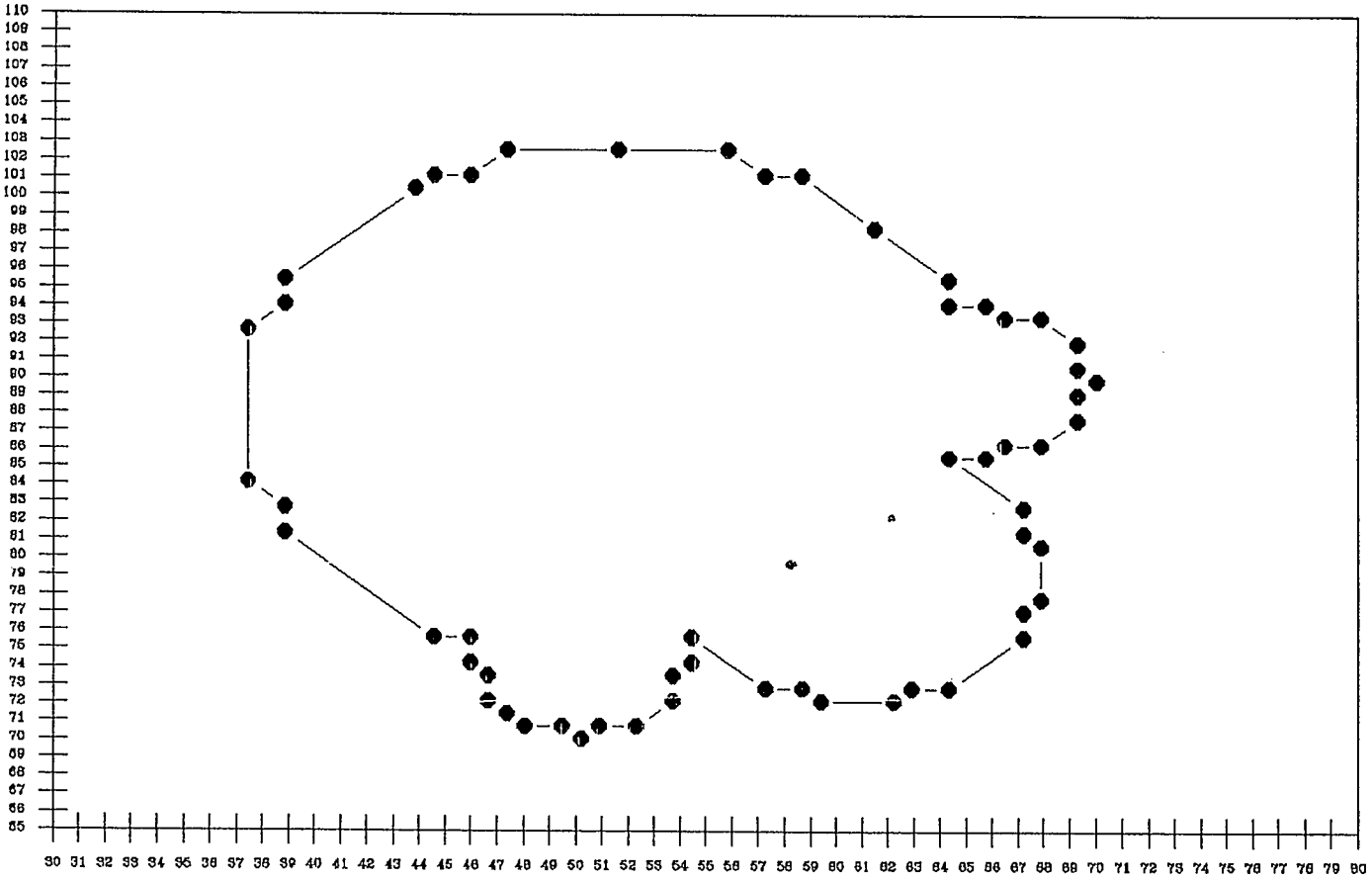


Fig. 5.36 Detected by Second Method (Rotation).

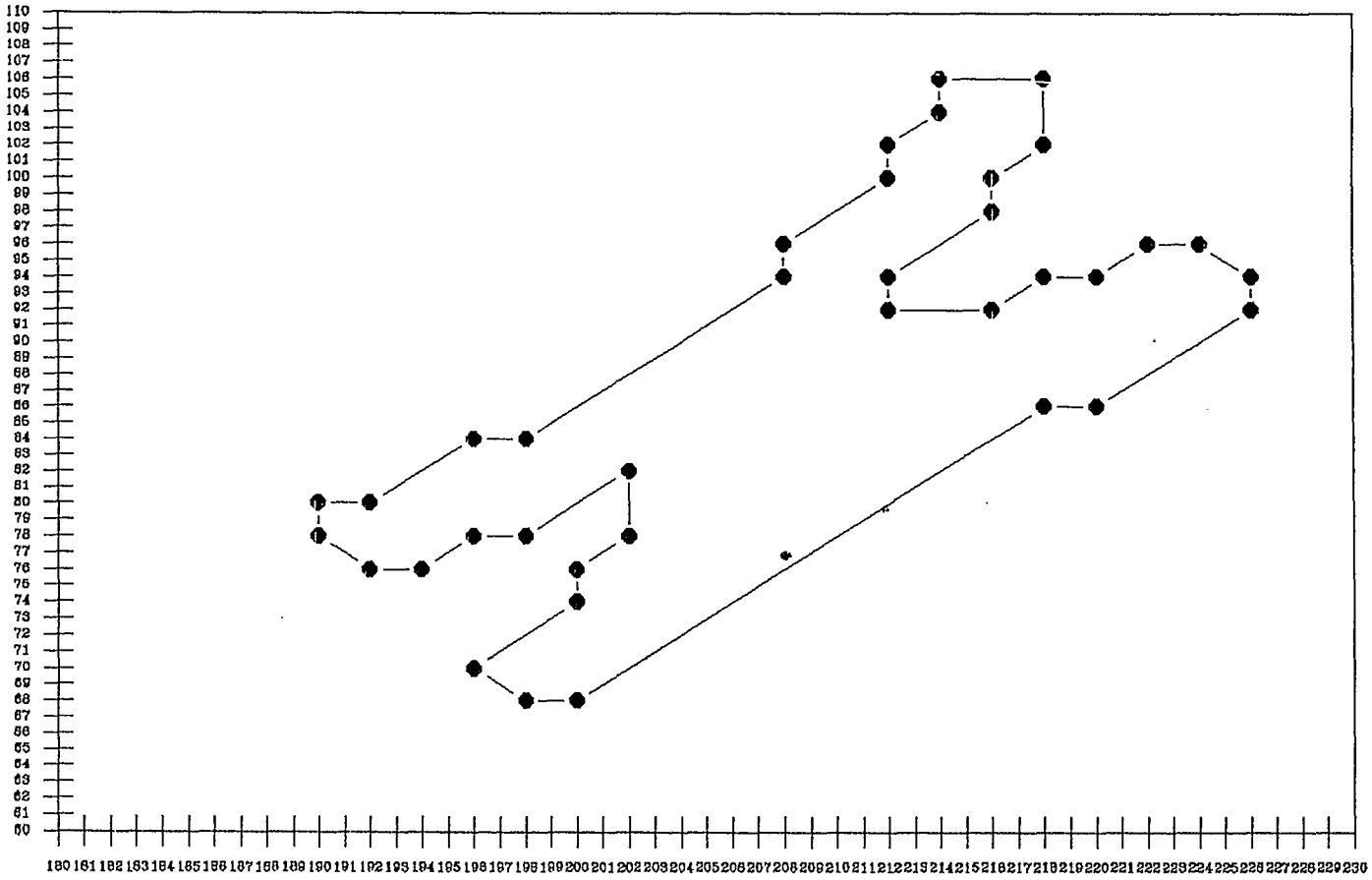


Fig. 5.37 Detected by Second Method (Scaling).

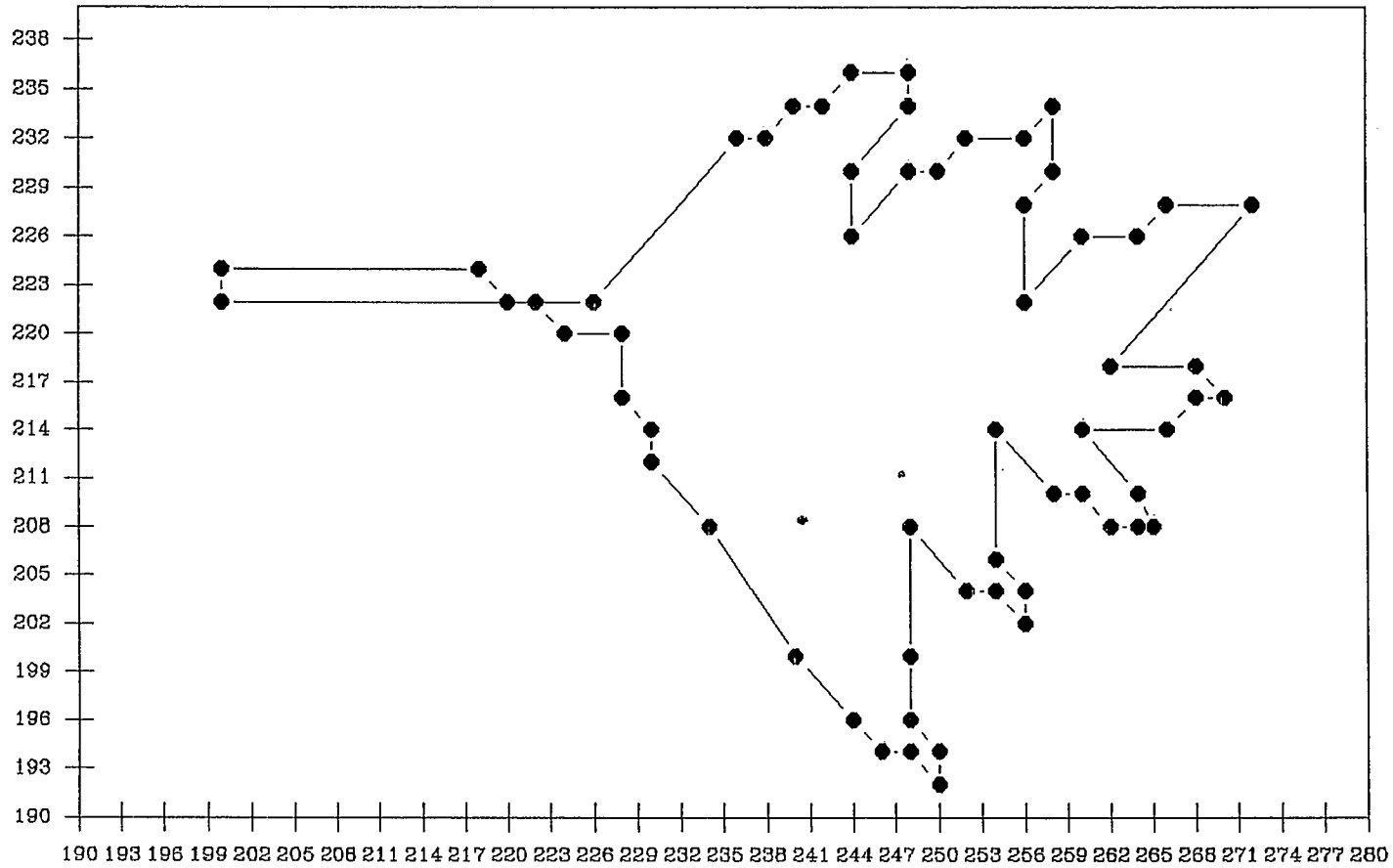


Fig. 5.38 Detected by Second Method (Scaling).

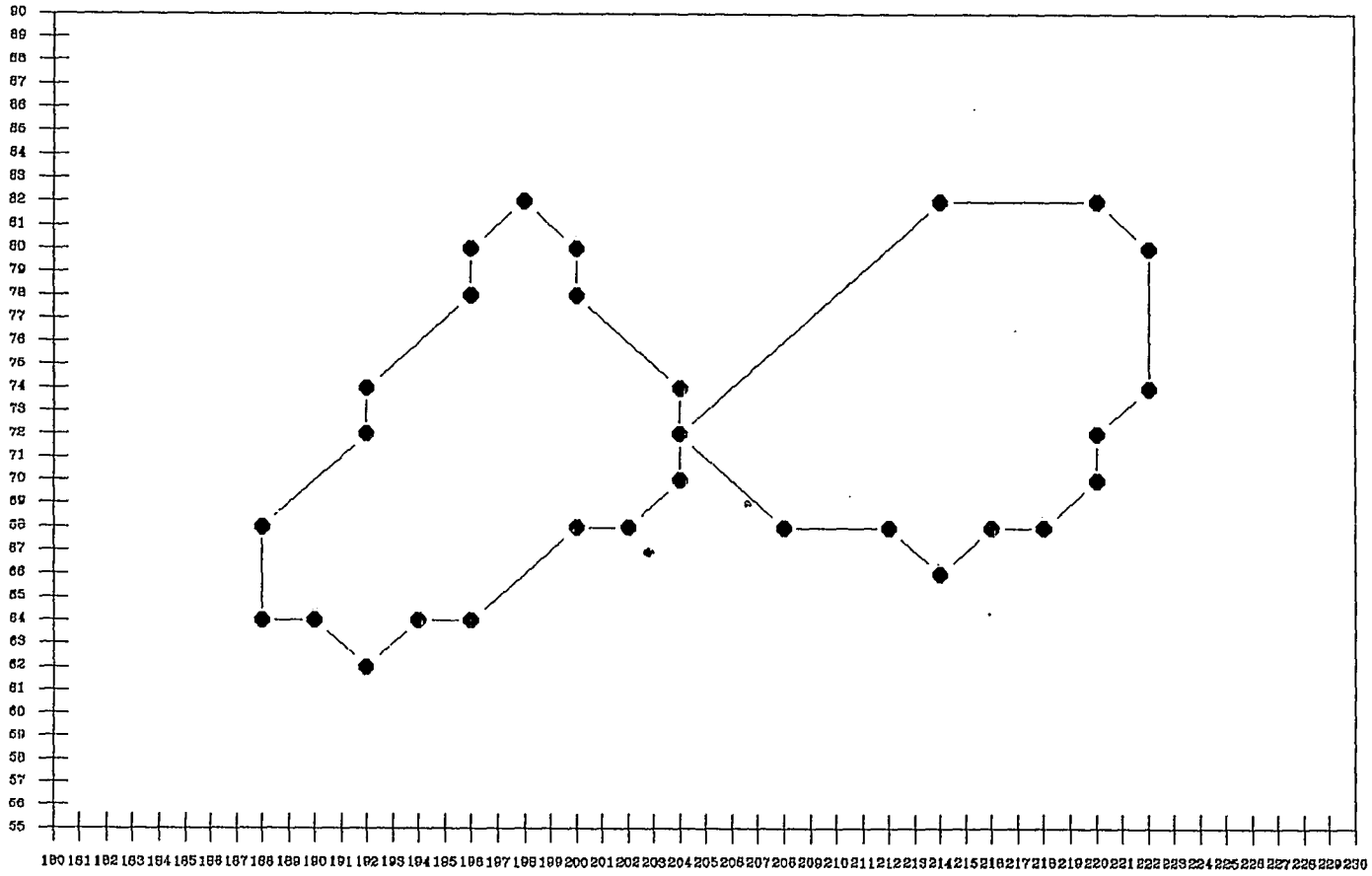


Fig. 5.39 Detected by Second Method (Scaling).

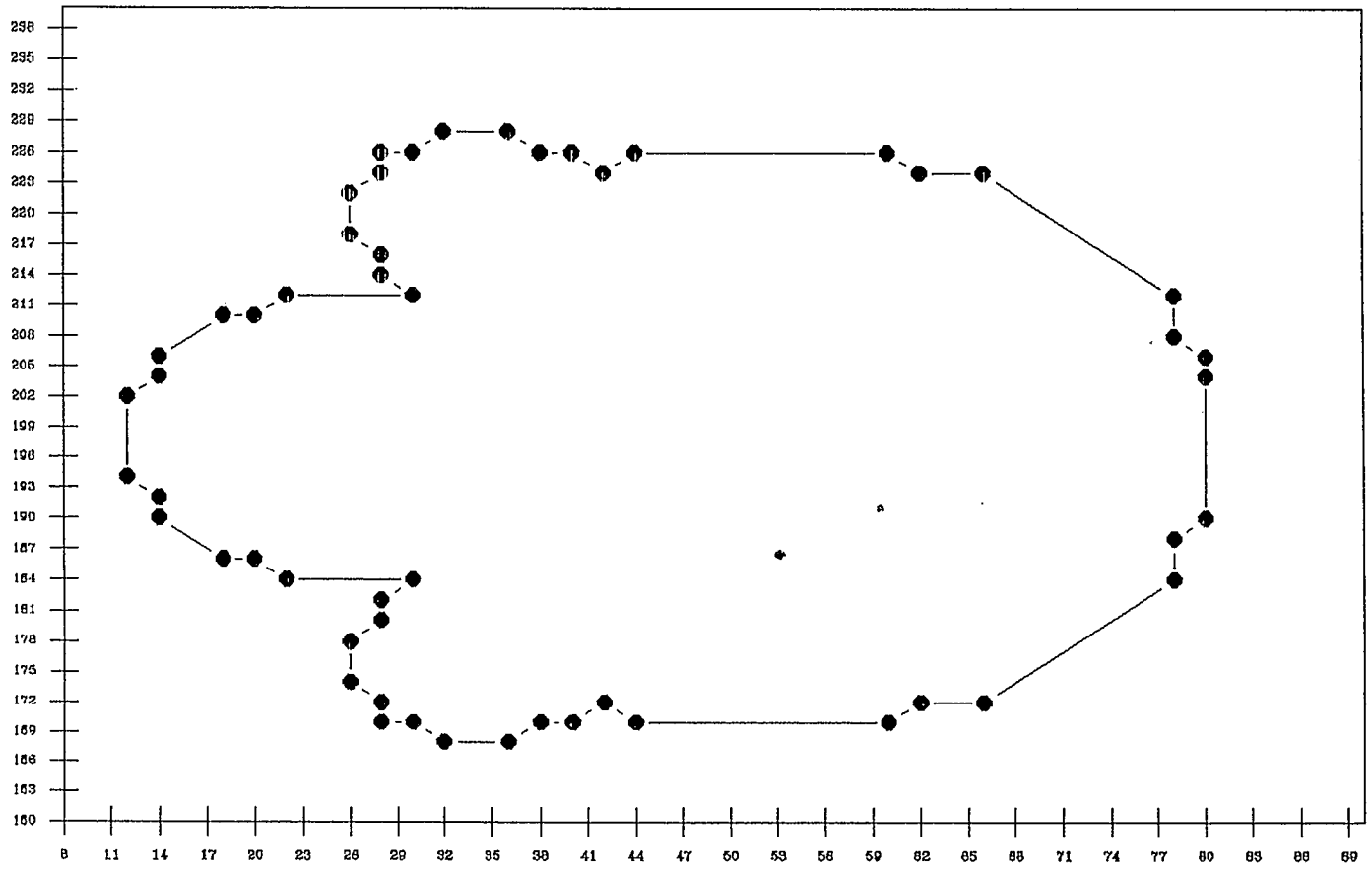


Fig. 5.40 Detected by Second Method (Scaling).

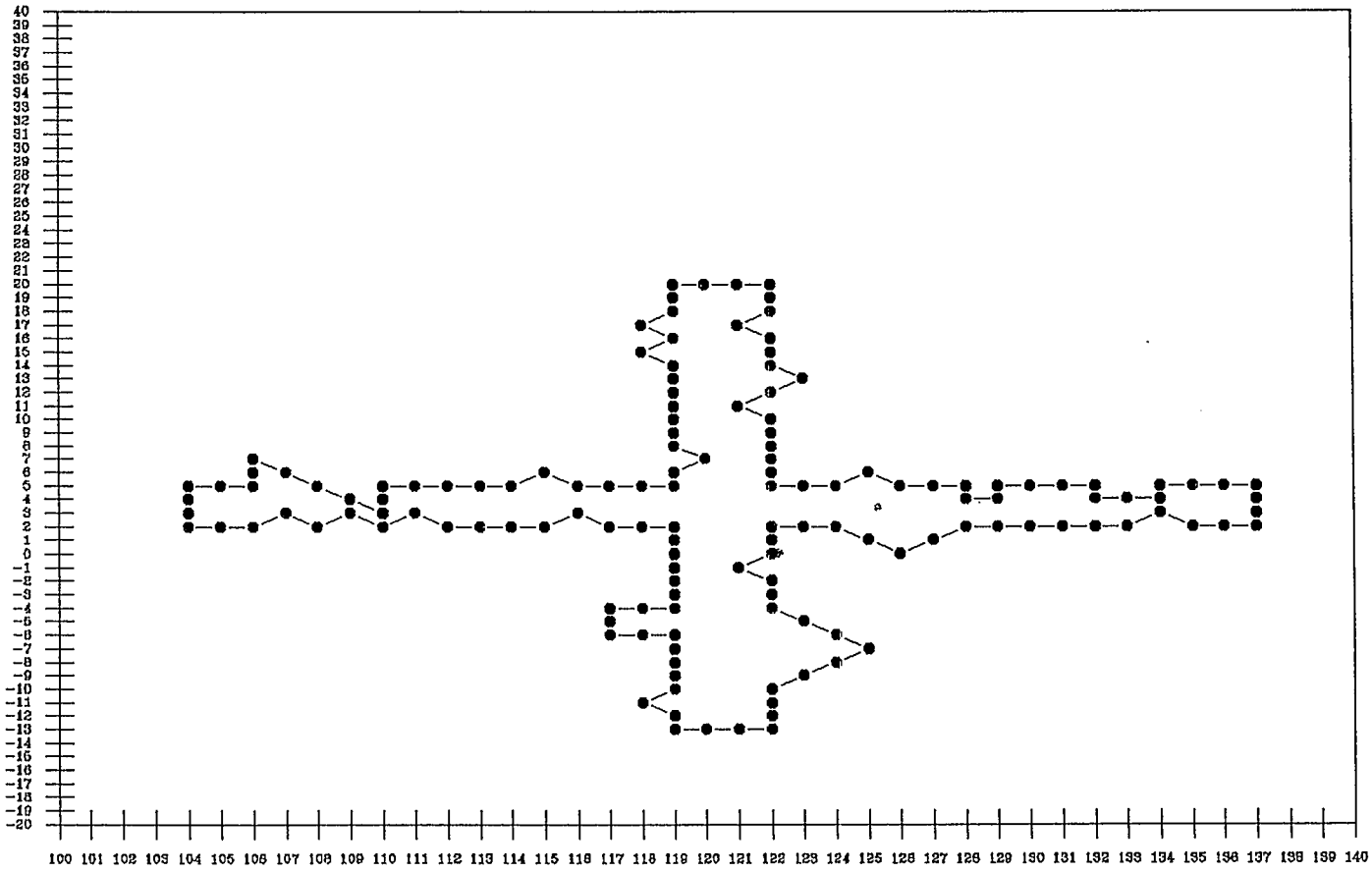


Fig. 41 The CROSS Contour.

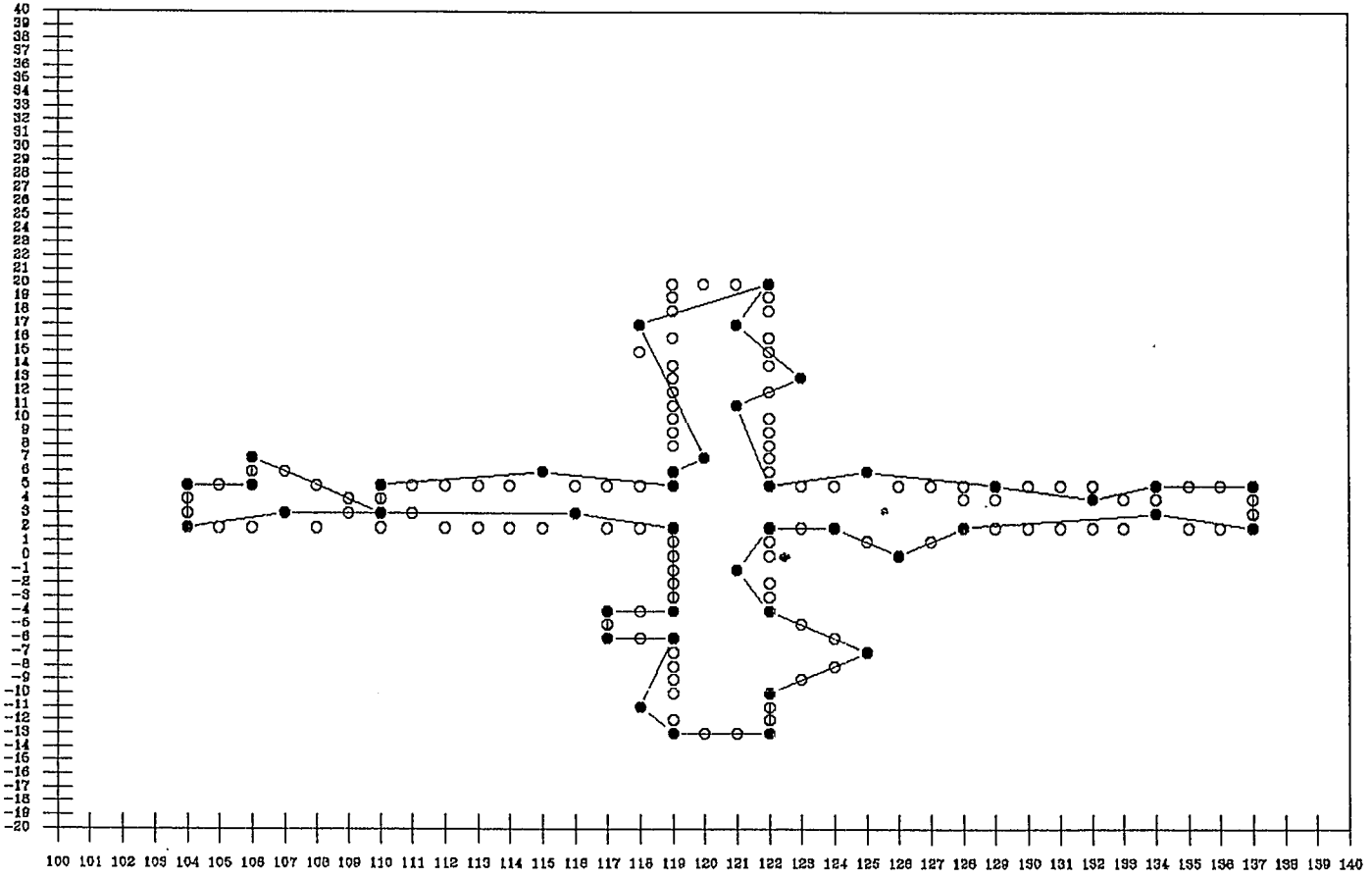


Fig. 5.42 Detected by Teh–Chin Algorithm.

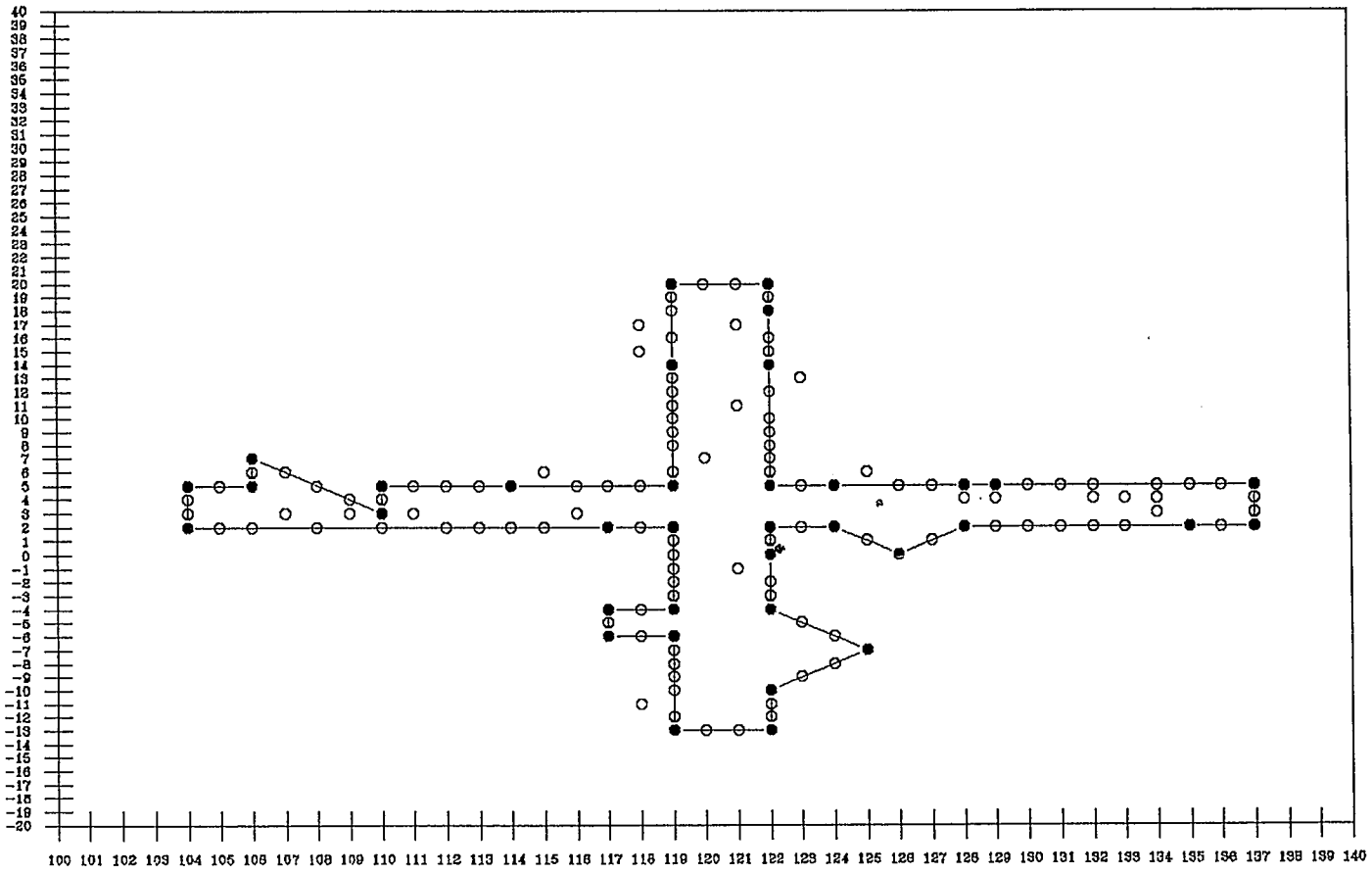
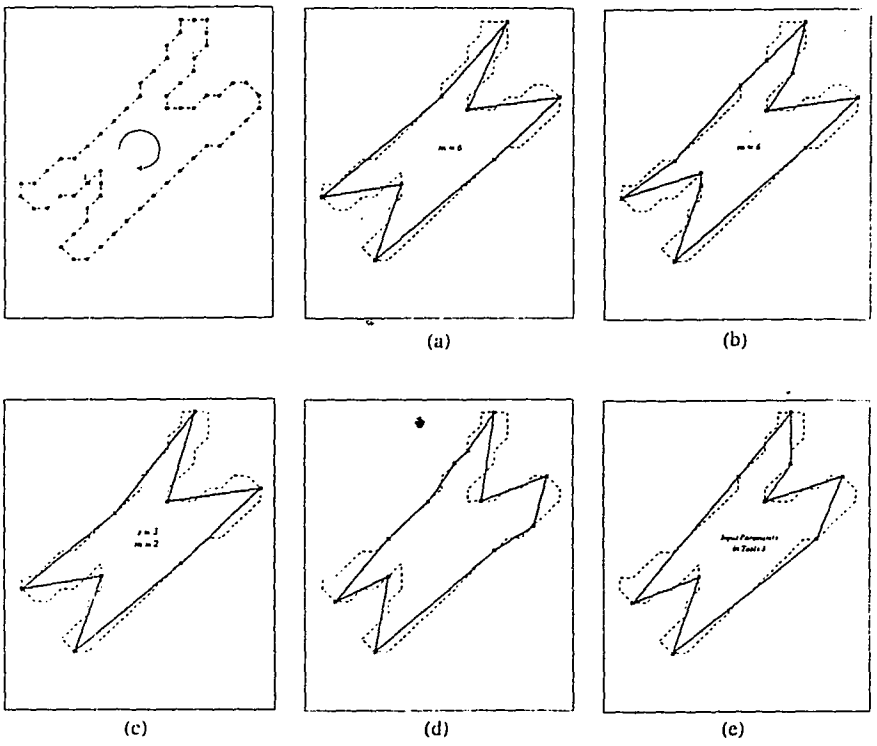


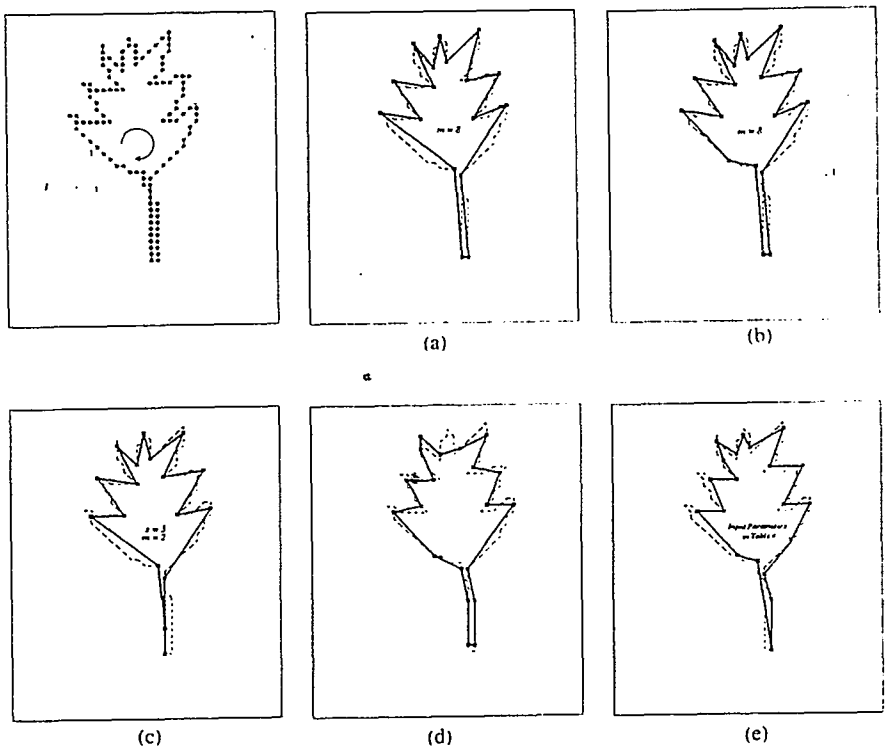
Fig. 5.43 Detected by First Method.

A-1 : The results of detecting CHROMOSOME contour by other algorithms.



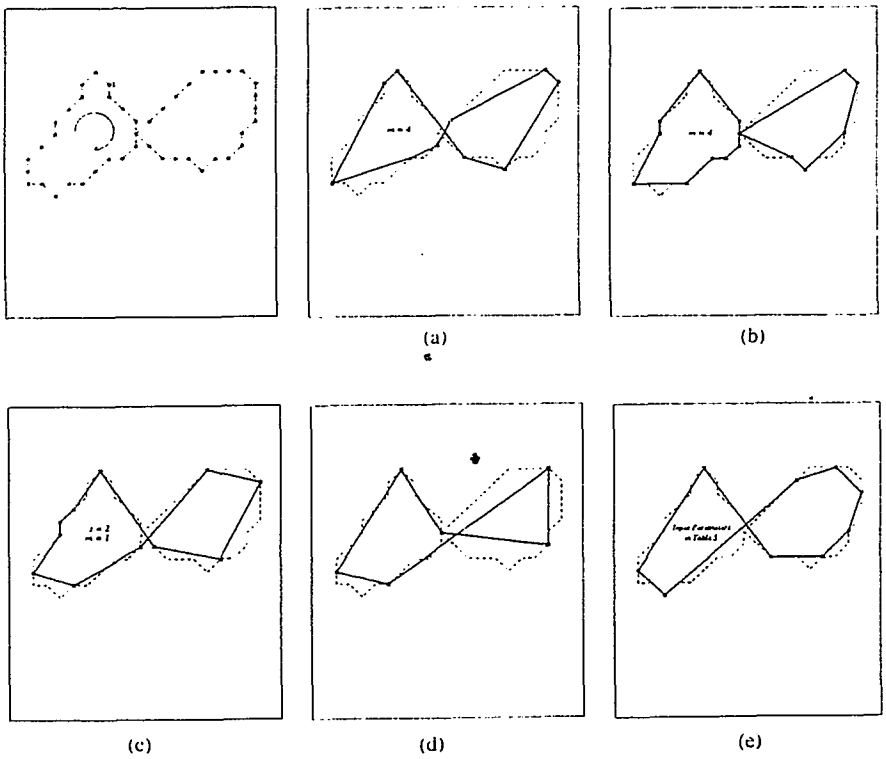
(a) Rosenfeld-Johnston algorithm. (b) Rosenfeld-Weszka algorithm. (c) Freeman-Davis algorithm. (d) Sankar-Sharma algorithm. (e) Anderson-Bezdek algorithm.

A-2 : The results of detecting LEAF contour by other algorithms.



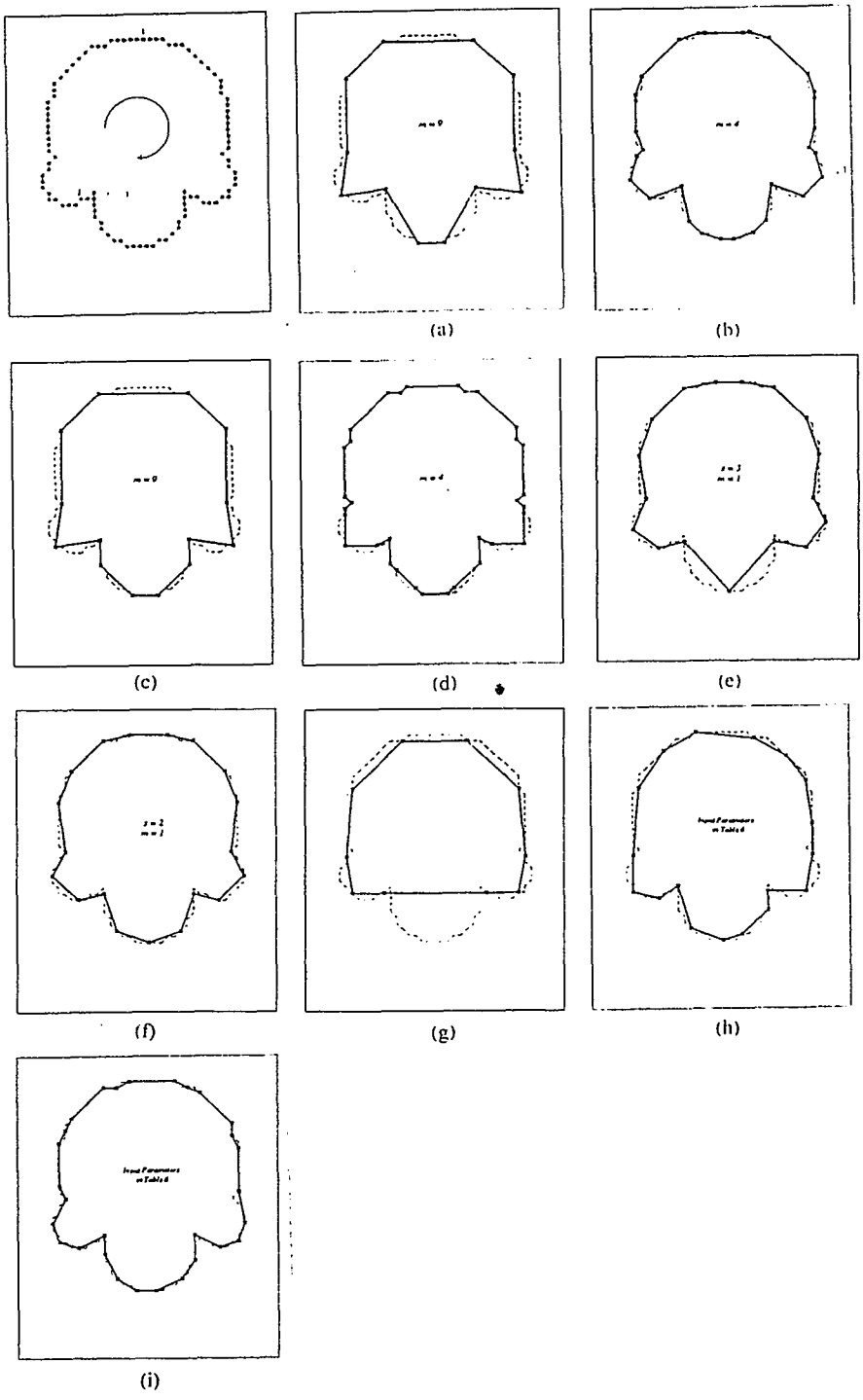
(a) Rosenfeld-Johnston algorithm. (b) Rosenfeld-Weszka algorithm. (c) Freeman-Davis algorithm. (d) Sankar-Sharma algorithm. (e) Anderson-Bezdek algorithm.

A-3 : The results of detecting EIGHT contour by other algorithms.



(a) Rosenfeld-Johnston algorithm. (b) Rosenfeld-Weszka algorithm. (c) Freeman-Davis algorithm. (d) Sankar-Sharma algorithm. (e) Anderson-Bezdek algorithm.

A-4 : The results of detecting SEMICIR contour by other algorithms.



(a) Rosenfeld-Johnston algorithm. (b) Rosenfeld-Johnston algorithm. (c) Rosenfeld-Weszka algorithm. (d) Rosenfeld-Weszka algorithm. (e) Freeman-Davis algorithm. (f) Freeman-Davis algorithm. (g) Sankar-Sharma algorithm. (h) Anderson-Bezdek algorithm. (i) Anderson-Bezdek algorithm.

APPENDIX B

Chain Code for Fig. 5.1					
55454	32011	01111	12112	12006	65655
60010	10765	55455	55555	55431	12122

Chain Code for Fig. 5.2					
33332	30700	00003	32307	00003	32322
26777	22212	76661	11116	66566	55000
10056	65655	00110	66565	65555	56667
66666	66664	22222	22222	23224	43433

Chain Code for Fig. 5.3					
76776	77007	10121	22234	44555	55654
55453	42211	21121			

Chain Code for Fig. 5.4					
00007	00777	77766	76666	66665	76766
56454	43436	66656	55454	44434	33232
22254	54434	23221	21322	22222	21221
11111	00100	00			

Bibliography

[1] I. M. Anderson and J. C. Bezdek,"Curvature and tangential deflection of discrete arcs: A theory based on the commutator of scatter matrix paris and its application to vertex detection in planar shape data," *IEEE Trans. Pattern Anal. Machine Intell.*,vol. PAMI-6, pp. 27-40, Jan. 1984.

[2] N. Ansari and E. J. Delp, "Partial shape recognition : A landmark-based approach," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. PAMI-12, No. 5, pp. 470-483 May 1990.

[3] N. Ansari and E. J. Delp, "A note on detecting dominant point detection," *Proc. Visual communication and image processing* IV, Philadephia, DA, pp. 821-832, 1989.

[4] F. Attneave, "Some informational aspects of visual perception," *Psychological review* vol. 61, 1954, 183-193.

[5] N. Ayache and O. D. Faugeras, "HYPER:A new approach for the recognition and positioning of two-dimensional objects, " *IEEE Trans. Pattern Anal. Machine. Intell.* vol. PAMI-8, pp. 44-54, 1986.

[6] W. H. Beyer, *CRC Standard Mathematical Tables*, Boca Raton, Florida: CRC Press, 28th ed., pp. 121, 1981.

[7] B. Bhanu and O. D. Faugeras,"Shape matching of two-dimension Objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 137-155, 1984.

[8] R. L. Burden and J. D.Faires, *Numerical Analysis,* third edition, N.Y : McGraw-Hill, 1982, pp. 184-190.

[9] L. S. Davis, "Understanding shape: Angles and sides," *IEEE Trans Comput.*, vol C-26, pp. 236-242, Mar. 1977.

[10] L. S. Davis and A. Rosenfeld, "Detection of step edges in noisy one-dimensional data,"*IEEE Trans. Comput.*, vol. C-21, pp. 677-715, July 1972.

[11] J. G. Dunham, "Optimum uniform piecewise linear approximation of planar," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 67-75, Jan. 1986.

[12] H. Freeman and L. S. Davis "A corner-finding algorithm for chain coded curves," *IEEE Trans. Comput.*, vol. C-26, pp. 297-303, Mar 1977.

[13] Gonzalez and Wintz, *Digital Image Processing*, second edition, N.Y : Addison Wesley, 1987, pp. 331-340.

[14] F. C. A. Groan and P. W. Verbeek, "Freeman-code probabilities of object boundary quantized contours," *Comput. Vision. Graphics Image Processing*, vol.7, pp. 183-192, 1972.

[15] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs : Prentice-Hall, 1989, pp. 119-120.

[16] M. W. Koch and R. L. Kashyap, "Using polygons to recognize and locate partially occluded objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 483-494, 1987.

[17 J. S. Milton and Jesse C. Arnold, *Probability and Statistics in the Engineering and Computing Sciences*, N.Y : McGraw-Hill, 1986, pp. 109-116.

[18] M. E. Mortenson, *Geometric Modeling*, N.Y : John Wiley, 1985, pp. 125-149.

[19] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York : McGraw-hill, 1987, 2nd Ed., pp 73-77.

[20] T. Pavlidis, "Algorithm for shape analysis and waveforms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 301-312, July 1980.

[21] T. Pavlidis and S. L. Horowitz, "Segmentation of Plane Curves," *IEEE Trans. Comput.*, c-23, pp. 860-870, 1974.

[22] Rogers and Adams, *Mathematical elements for computer graphics*, second edition, N.Y : McGraw-Hill, 1989, pp. 461-465.

[23] A. Rosenfeld and E. Johnston, "Angle detection on digital curves," *IEEE Trans. Comput.*, vol. c-22, pp.875-878, Sept. 1973.

[24] A. Rosenfeld and A. C. Kak, *Digital picture processing*, second edition, N.Y : Academic Press, 1982, pp. 1-9.

[25] A. Rosenfeld and M. Thurston, "Edge and curve detection for digital scene analysis, "*IEEE Trans. Comput.*, vol. C-20, pp. 562-569, May 1971.

[26] A. Rosenfeld, M. Thurston. and Y. H. Lee. "Edge and curve detection : Further experiments," *IEEE Trans. Comput.*, vol. C-21, pp. 677-715. July 1972.

[27] A. Rosenfeld and J. S. Weszka, "An improved method of angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-24, pp. 940-941, Sept. 1975.

[28] P. V. Sankar and C. V. Sharma, "A parallel procedure for the detection of dominant points on a digital curve," *Comput. Graphics Image Processing*, vol. 7, pp. 403-412, 1978.

[29] C. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Trans. on Pattern and Machine Intell.*, vol. 11, No 8, August 1989.

[30] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. on Pattern Anal and Machine Intell.*, PAMI-8, No.2, pp. 147-163, March 1986.

[31] L. D. Wu, "A piecewise linear approximation based on a statistical model," *IEEE Trans.Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 41-45, Jan. 1984.